Topic Modeling with BERTopic

Objective

- The main objective of this project is to be able to cluster the news article documents focused on Technology, Health and generate the topics for a given article. In order to tackle the problem, a Neural topic modeling(BERTopic [link text](#)) with a class-based TF-IDF procedure is implemented.

The main section of this notebook organize as follows:

- Load Technology and Health Sample Data.
- Apply the all-mpnet-base-v2 model which provides the best quality for sentence embedding.
- Cluster document using HDBSCAN clustering algorithm.
- Apply C-TF-IDF on each cluster and generate Topics for each cluster Dynamic Topics Over period of time.

Import necessary modules

```
import pandas as pd
pd.set_option('max_colwidth',150)
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime as dt
from string import punctuation
import re
import os
from sklearn.feature_extraction.text import CountVectorizer
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all" # allow multiple outputs in a cell
import warnings
import pandas as pd
# pd.options.plotting.backend = "plotly"
# warnings.filterwarnings("ignore")
%matplotlib inline
```

Download and Extract the Datasets

✓ 0s completed at 1:22 PM

```
pathdir = "/content/data"
```

```
def download_dataset():

  if not os.path.isfile('all-the-news-2-1.zip?dl=0'):
    ! wget https://www.dropbox.com/s/cn2utnr5ipathhh/all-the-news-2-1.zip?dl=0
    # !gdown --id 1LEbse_I7hrbPSwgSFGGFDrdlqbG99LUb

    # Downloading Annotated Corpus for Named Entity Recognition dataset
    !gdown https://drive.google.com/uc?id=13y8JNgL5TQ4x-yufpBOv3QBsEiE051sE

  if not os.path.exists(pathdir):
    # !unzip reuters21578-20211110T171613Z-001.zip
    # Make a data folder to store the data
    !mkdir data

    !unzip /content/all-the-news-2-1.zip?dl=0 -d ./data/

    !mv /content/ner.csv ./data

    !rm /content/all-the-news-2-1.zip?dl=0
```

```
download_dataset()
```

Load Data

```
#specify the path to data location

filepath = '/content/data/all-the-news-2-1.csv'
# data = pd.read_csv(filepath, encoding = "ISO-8859-1")
data = pd.read_csv(filepath, encoding = "utf-8")
```

```
#Verify that the data is loaded correctly
data.head(3)
```

	date	year	month	day	author	title	article
							This post is part of Polyarchy,

0	2016-12-09 18:31:00	2016	12.0	9	Lee Drutman	We should take concerns about the health of liberal democracy seriously	an independent blog produced by the political reform program at New America, a Washington think tank devoted to de...	https://www.vo /201 /democra
						Colts GM	The Indianapolis Colts made	

```
#totally the data have 2,688,878 rows and 10 columns
data.shape
```

```
(2688878, 10)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2688878 entries, 0 to 2688877
Data columns (total 10 columns):
 #   Column       Dtype
---  ------       -----
 0   date         object
 1   year         int64
 2   month        float64
 3   day          int64
 4   author       object
 5   title        object
 6   article      object
 7   url          object
 8   section      object
 9   publication  object
dtypes: float64(1), int64(2), object(7)
memory usage: 205.1+ MB
```

```
data.isnull().sum()
```

```
date                 0
year                 0
month                0
day                  0
author         1021101
title               37
article         104713
url              12577
```

```
section         912273
publication      12577
dtype: int64
```

```
def filter_section(section):

  if str(section).lower().startswith('tech') :
    return 'technology'
  elif str(section).lower().startswith('health'):
    return 'health'

  return 'other'
```

```
data['tech_health_tag'] = data['section'].apply(filter_section)
```

```
data['tech_health_tag'].value_counts()
```

```
other         2562768
health          65261
technology      60849
Name: tech_health_tag, dtype: int64
```

Load the data which focus only on Health and Technology Section

```
data_tech_health = data[(data['tech_health_tag']=='technology') |  (data['tech_hea
```

```
data_tech_health = data_tech_health.reset_index(drop=True)
```

```
data_tech_health.head(3)
```

```
# save the tech and health data and delete all-the-news data inorder to save memor
import gc

data_tech_health.to_csv('/content/data/tech_health_data.csv', index=False)
del data
gc.collect()
```

```
4
```

```
data_tech_health.shape
```

```
(126110, 11)
```

```
data_tech_health.info()
```

```
data_tech_health.isnull().sum()
```

```
data_tech_health['tech_health_tag'].value_counts()
```

```
health        65261
technology    60849
Name: tech_health_tag, dtype: int64
```

```
data_tech_health['publication'].unique()
```

```
array(['Vice', 'Reuters', 'The Verge', 'People', 'Economist', 'CNN',
       'Gizmodo', 'CNBC', 'Fox News', 'The New York Times'], dtype=object)
```

```
data_tech_health['year'].value_counts().sort_index()
```

```
2016    24470
2017    28697
2018    24770
2019    22961
2020    25212
Name: year, dtype: int64
```

```
data_tech_health['year'].unique()
```

```
array([2018, 2019, 2017, 2016, 2020])
```

Data Cleaning

```
def processed_text_article(df):
    special_char = list(punctuation)
    for e in ['.','?']:
        special_char.remove(e)
    special_char.append("\n+")
    special_char.append("\s+")
    special_char.append("said")
    special_char.append("told")
    special_char.append("like")
    special_char.append("just")


    def deep_clean(text_str):
```

```
    def deep_clean(text_str):
      text_str = str(text_str)
      text_str =text_str.lower().strip()
      text_str = re.sub('<[^>]*>', '', text_str)
      text_str = re.sub("\'", "", text_str)  # remove single quotes
      text_str = re.sub('\S*@\S*\s?', '', text_str)  # remove emails
      for char in special_char:
          text_str = text_str.replace(char, '')
      return text_str

    df['article'] = df['article'].apply(deep_clean)
    df['title'] = df['title'].apply(deep_clean)
    return df

def clean_dataFrame(df):
  missing_cols = df.isnull().sum()
  drop_missing_cols = missing_cols[(missing_cols > len(df)/20)].sort_values()
  df = df.drop(drop_missing_cols.index, axis=1)
  df['date'] = pd.to_datetime(df['date'])
  df = df.dropna()
  #reset index
  df = df.reset_index(drop=True)
  # make all columns lower_case
  df.columns = df.columns.str.lower()
  df = processed_text_article(df)
  return df
```

```
data_tech_health = clean_dataFrame(data_tech_health)
```

```
data_tech_health.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 125948 entries, 0 to 125947
Data columns (total 10 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   date             125948 non-null  datetime64[ns]
 1   year             125948 non-null  int64
 2   month            125948 non-null  float64
 3   day              125948 non-null  int64
 4   title            125948 non-null  object
 5   article          125948 non-null  object
 6   url              125948 non-null  object
 7   section          125948 non-null  object
 8   publication      125948 non-null  object
 9   tech_health_tag  125948 non-null  object
dtypes: datetime64[ns](1), float64(1), int64(2), object(6)
memory usage: 9.6+ MB
```

```
data_tech_health.isnull().sum()
```

```
date              0
year              0
month             0
day               0
title             0
article           0
url               0
section           0
publication       0
tech_health_tag   0
dtype: int64
```

```
# remove stopwords
# Import stopwords with nltk.
import nltk
nltk.download("stopwords")
from nltk.corpus import stopwords
# STOPWORDS = set(stopwords.words('english'))
stop = stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
print(stop)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're"
```

```
# Remove stop words
data_tech_health['article'] = data_tech_health['article'].apply(lambda x: ' '.join
```

Modeling

- Apply Bertopic language model for document clustering and Topic generation. more information can be get [on this paper.]()

```
!pip install bertopic
!pip install bertopic[visualization]
```

```
from bertopic import BERTopic
```

Cluster Technology Article Sample Data and Generate Topics

Topics

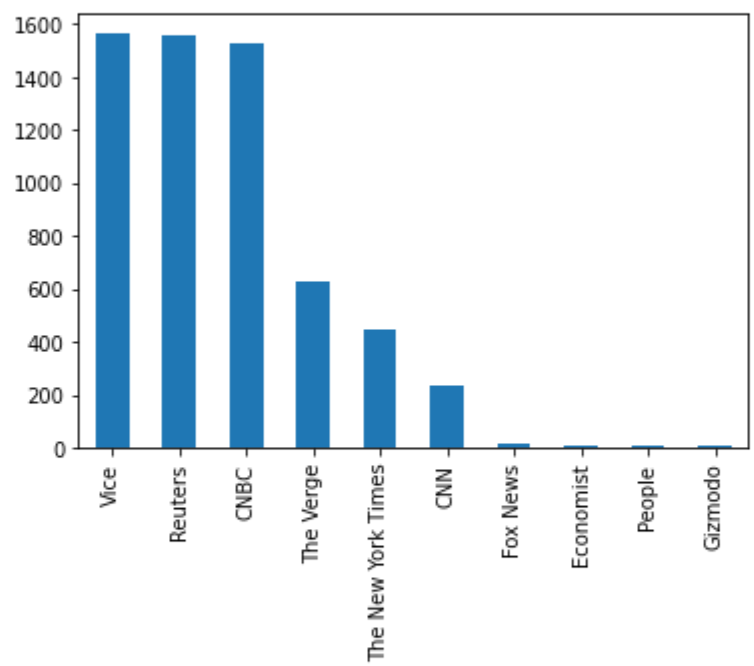
- Load Technology Article Sample Data
- Apply the all-mpnet-base-v2 model which provides the best quality for sentence embedding.
- Cluster document using HDBSCAN clustering algorithm
- Apply C-TF-IDF on each cluster and generate Topics for each cluster
- Dynamic Topics Over period of time

```
# load 6000 data which
data_tech = data_tech_health[data_tech_health['tech_health_tag']=='technology']
data_tech_sample = data_tech.sample(n=6000, random_state=42, ignore_index=True)
```

```
data_tech_sample.head()
```

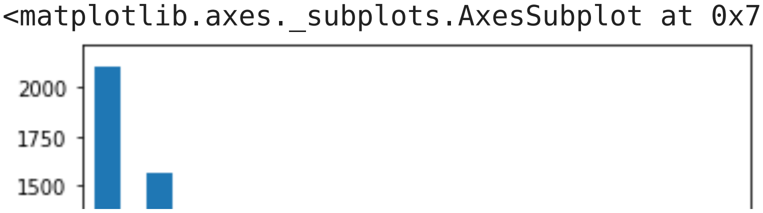
```
data_tech_sample['publication'].value_counts().plot.bar()
```

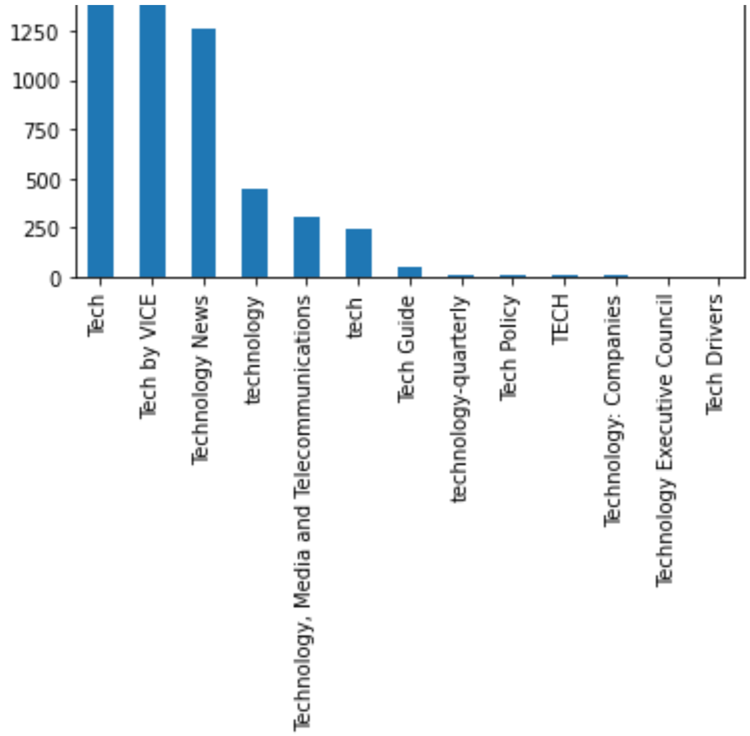
<matplotlib.axes._subplots.AxesSubplot at 0x7f662a67df90>




```
data_tech_sample['section'].value_counts().plot.bar()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f654803c990>






```
# SentenceTransformers is a Python framework for state-of-the-art sentence, text a

# !pip install -U sentence-transformers
```

```
# from sentence_transformers import SentenceTransformer

# #convert to list
# docs = data_tech_sample.article.to_list()

# # Prepare embeddings
# #By default, input text longer than 128 word pieces is truncated.
# sentence_model = SentenceTransformer("all-mpnet-base-v2")
# embeddings = sentence_model.encode(docs, show_progress_bar=False)
```

```
# embeddings.shape
```

Train BERTopic Model

```
# Train our topic model using our pre-trained sentence-transformers embeddings
# nr_topics="auto" merge similar topics

model_tech = BERTopic(verbose=True)

#convert to list
```

```
docs_tech = data_tech_sample.article.to_list()

topics, probabilities = model_tech.fit_transform(docs_tech)
```

Select Top Topics

```
# After training the model, you can access the size of topics in descending order
model_tech.get_topic_freq()
```

	Topic	Count
0	-1	1720
1	0	194
2	1	180
3	2	159
4	3	122
...	...	...
108	107	11
109	108	11
110	109	11
111	110	11
112	111	10

113 rows × 2 columns

Note:

Topic -1 is the largest and it refers to outliers tweets that do not assign to any topics generated. In this case, we will ignore Topic -1.

Select One Topic

- You can select a specific topic and get the top n words for that topic and their c-TF-IDF scores

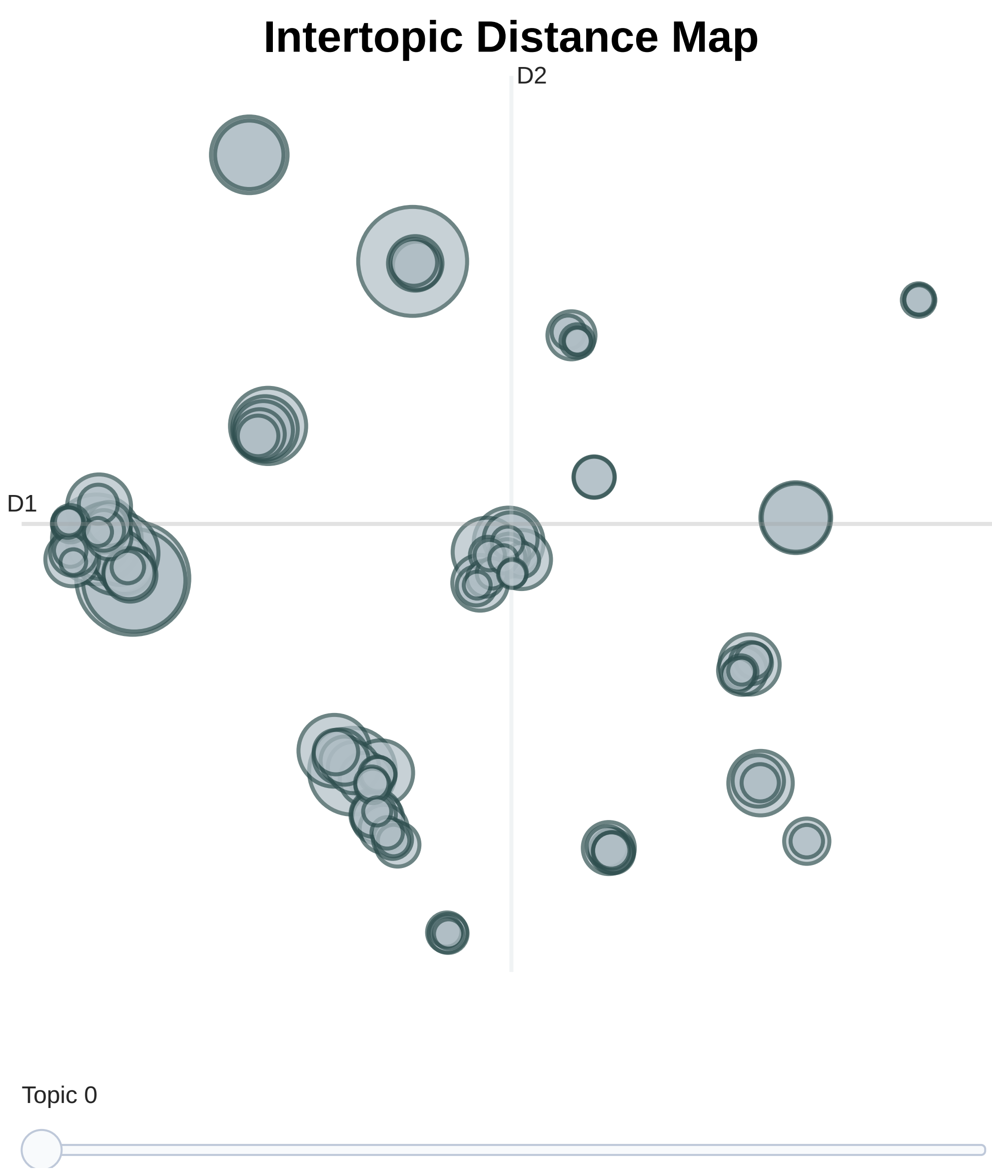
```
model_tech.get_topic(0)
```

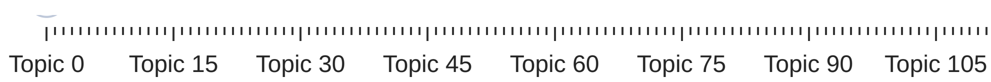
```
[('game', 0.033701591036420 71),
 ('games', 0.024838298979861582),
 ('players', 0.014479392699899342),
 ('nintendo', 0.013355242445978 29),
 ('console', 0.010268288463360848),
 ('video', 0.009656740420662325),
 ('play', 0.009459550418965 96),
 ('gaming', 0.008760919110248937),
 ('mario', 0.007855619706405335),
 ('esports', 0.006279976645306127)]
```

Visualize Topics

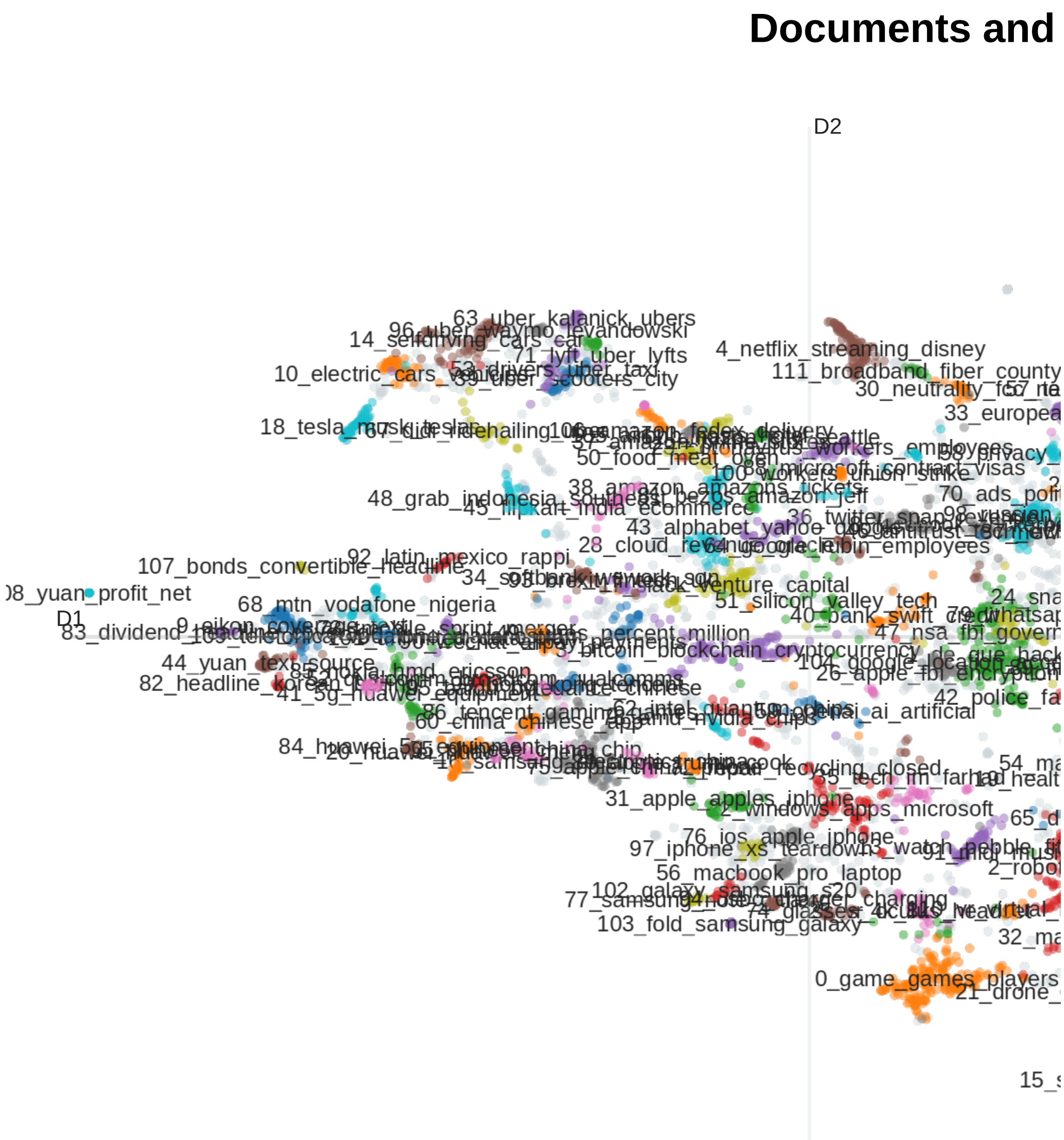
Visualize topics generated with their sizes and corresponding words

```
model_tech.visualize_topics()
```

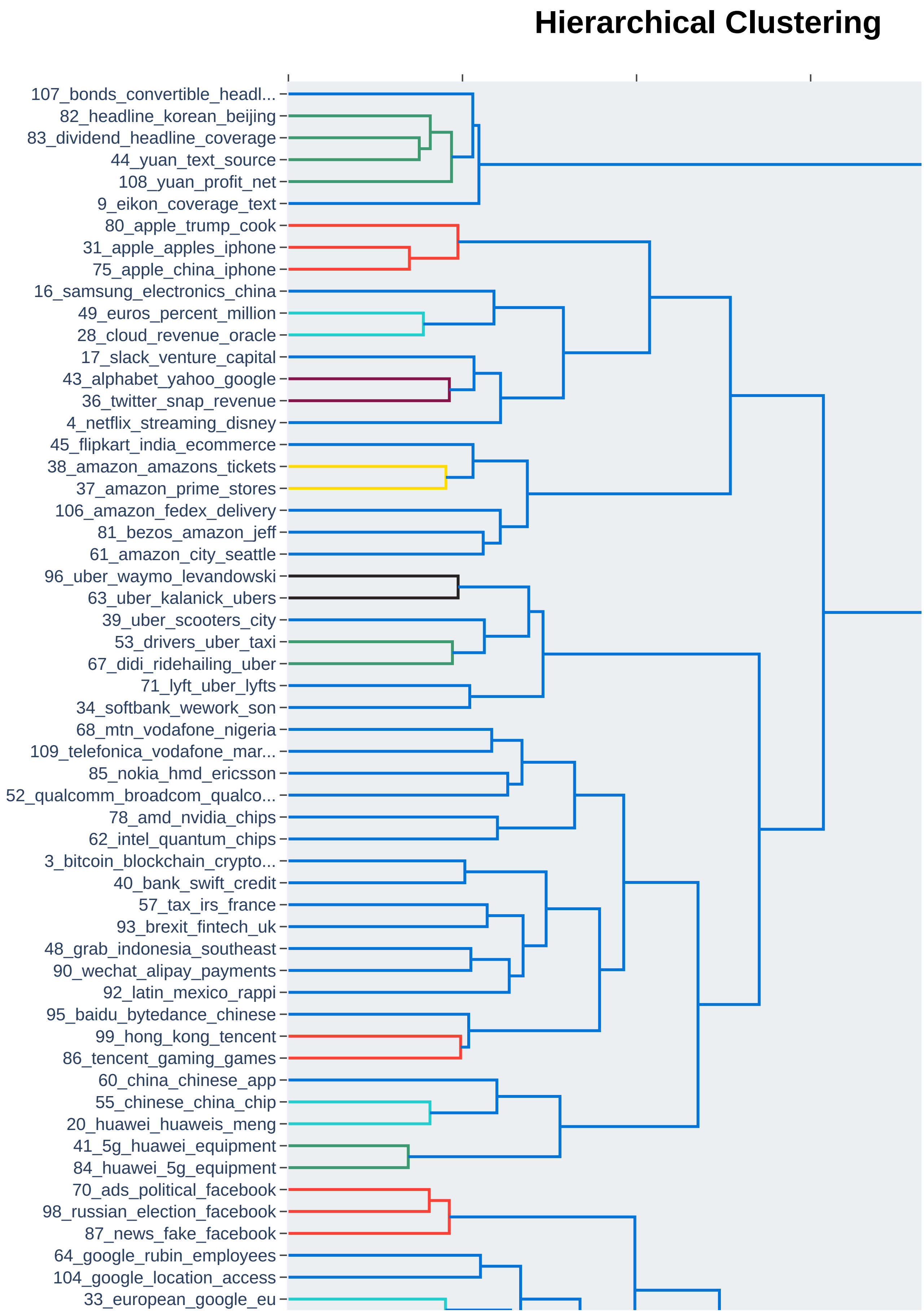


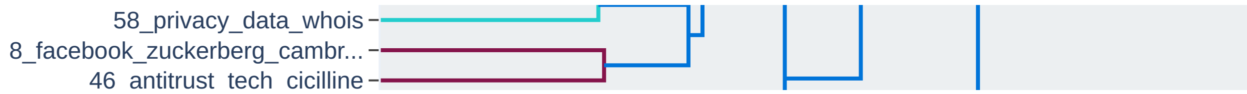



```
model_tech.visualize_documents(docs_tech)
```




```
model_tech.visualize_hierarchy()
```

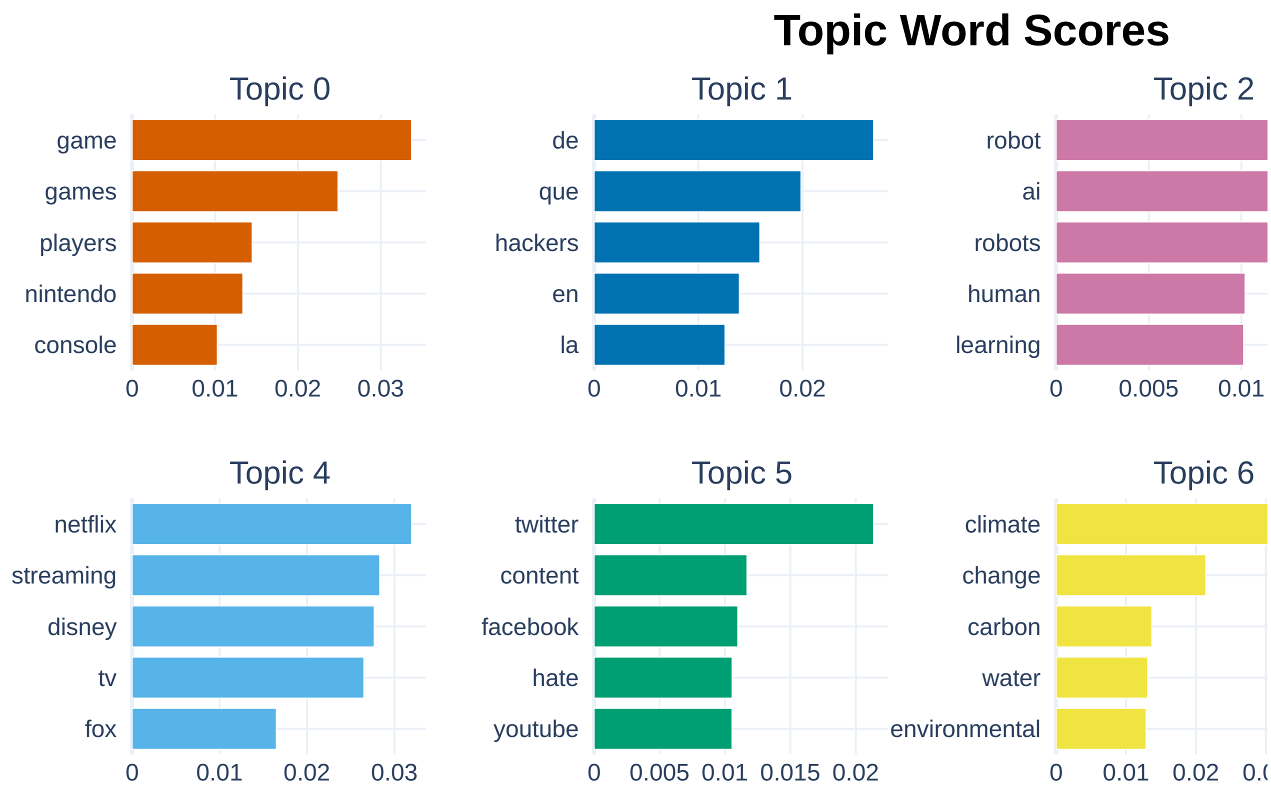

Hierarchical Clustering
107_bonds_convertible_headl...
82_headline_korean_beijing
83_dividend_headline_coverage
44_yuan_text_source
108_yuan_profit_net
9_eikon_coverage_text
80_apple_trump_cook
31_apple_apples_iphone
75_apple_china_iphone
16_samsung_electronics_china
49_euros_percent_million
28_cloud_revenue_oracle
17_slack_venture_capital
43_alphabet_yahoo_google
36_twitter_snap_revenue
4_netflix_streaming_disney
45_flipkart_india_ecommerce
38_amazon_amazons_tickets
37_amazon_prime_stores
106_amazon_fedex_delivery
81_bezos_amazon_jeff
61_amazon_city_seattle
96_uber_waymo_levandowski
63_uber_kalanick_ubers
39_uber_scooters_city
53_drivers_uber_taxi
67_didi_ridehailing_uber
71_lyft_uber_lyfts
34_softbank_wework_son
68_mtn_vodafone_nigeria
109_telefonica_vodafone_mar...
85_nokia_hmd_ericsson
52_qualcomm_broadcom_qualco...
78_amd_nvidia_chips
62_intel_quantum_chips
3_bitcoin_blockchain_crypto...
40_bank_swift_credit
57_tax_irs_france
93_brexit_fintech_uk
48_grab_indonesia_southeast
90_wechat_alipay_payments
92_latin_mexico_rappi
95_baidu_bytedance_chinese
99_hong_kong_tencent
86_tencent_gaming_games
60_china_chinese_app
55_chinese_china_chip
20_huawei_huaweis_meng
41_5g_huawei_equipment
84_huawei_5g_equipment
70_ads_political_facebook
98_russian_election_facebook
87_news_fake_facebook
64_google_rubin_employees
104_google_location_access
33_european_google_eu



Visualize Terms

The visualize_barchart method will show the selected terms for a few topics by creating bar charts out of the c-TF-IDF scores.

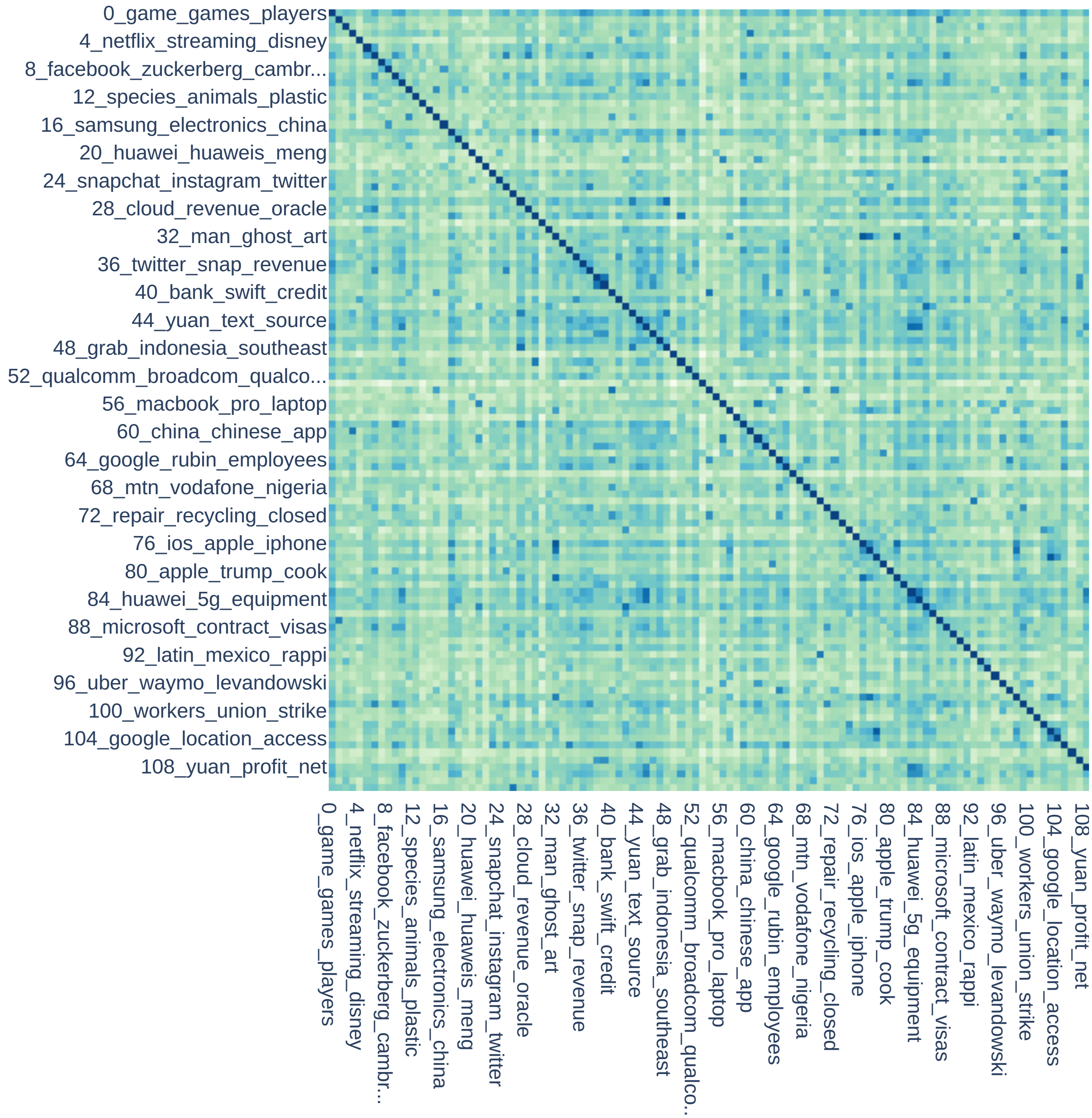
```
model_tech.visualize_barchart()
```



Visualize Topic Similarity

```
model_tech.visualize_heatmap()
```

Similarity Matrix



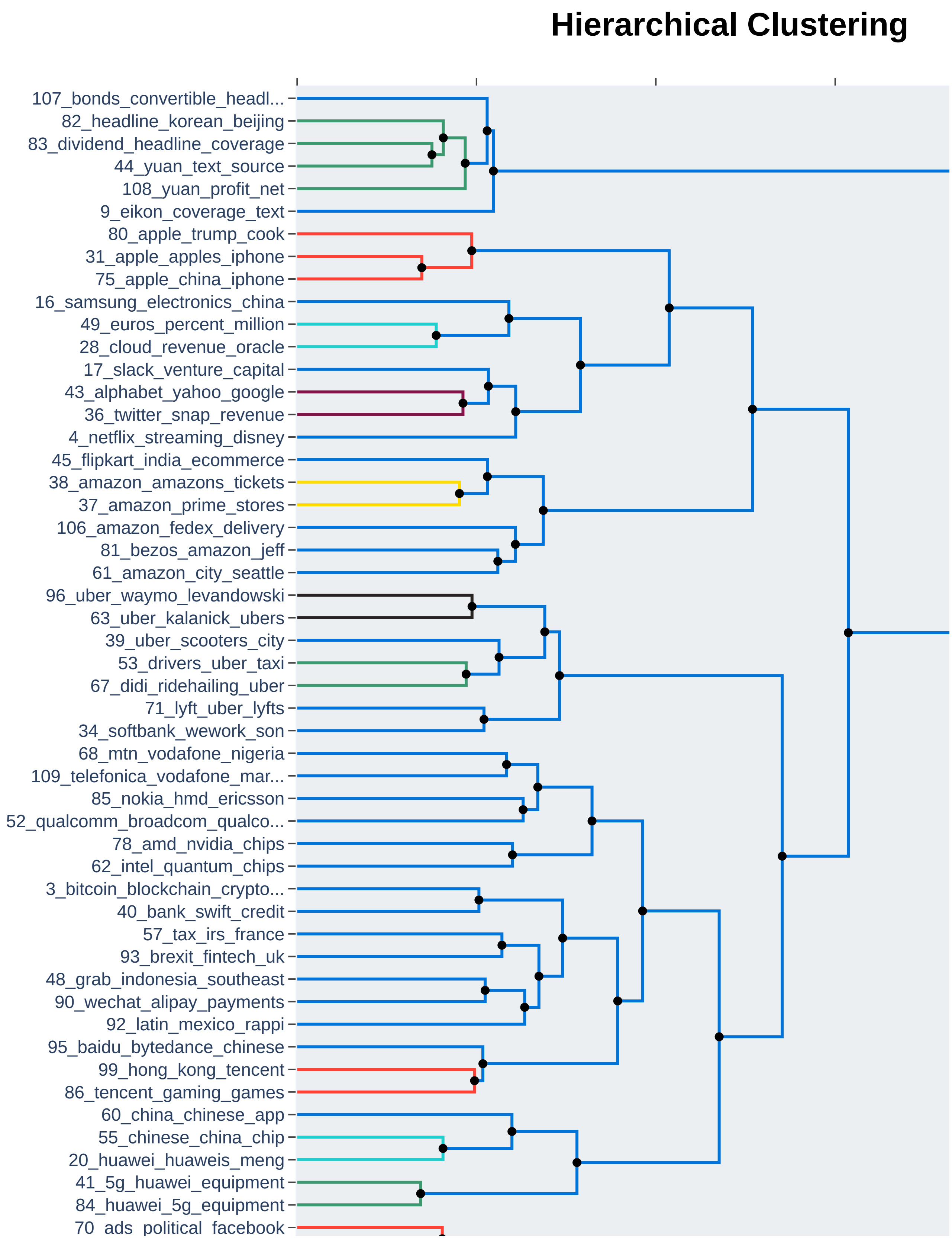
Hirarical Topics

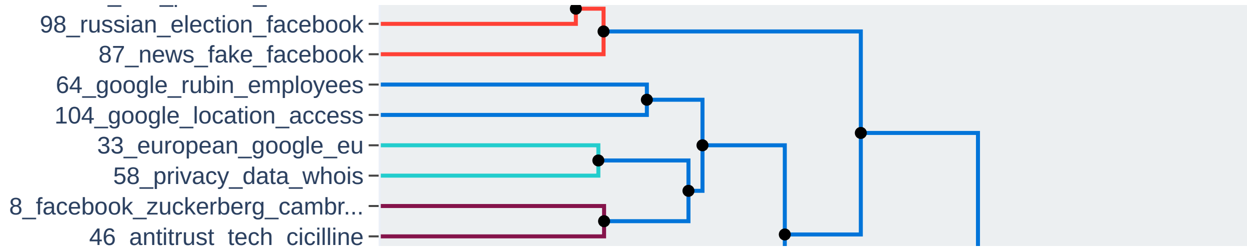
```
hierarchical_topics = model_tech.hierarchical_topics(docs_tech)
```

```
100%|██████████| 111/111 [00:01<00:00, 89.02it/s]
```

```
hierarchical_topics
```

```
model_tech.visualize_hierarchy(hierarchical_topics=hierarchical_topics)
```






```
tree = model_tech.get_topic_tree(hierarchical_topics)
print(tree)
```

```
.
├─coverage_text_source_ltd_million
│    ├─headline_text_coverage_yuan_beijing
│    │    ├─■──bonds_convertible_headline_coverage_wonshare ── Topic: 107
│    │    └─yuan_headline_text_coverage_beijing
│    │         ├─text_headline_coverage_source_beijing
│    │         │    ├─■──headline_korean_beijing_coverage_text ── Topic: 82
│    │         │    └─text_yuan_coverage_source_headline
│    │         │         ├─■──dividend_headline_coverage_text_beijing ── Topi
│    │         │         └─■──yuan_text_source_coverage_says ── Topic: 44
│    │         └─■──yuan_profit_net_fy_q1 ── Topic: 108
│    └─■──eikon_coverage_text_million_source ── Topic: 9
└─new_company_one_would_also
     ├─billion_company_uber_million_year
     │    ├─amazon_billion_apple_revenue_percent
     │    │    ├─apple_revenue_billion_quarter_percent
     │    │    │    ├─apple_apples_iphone_cook_china
     │    │    │    │    ├─■──apple_trump_cook_manufacturing_united ── Topic:
     │    │    │    │    └─apple_iphone_apples_cook_sales
     │    │    │    │         ├─■──apple_apples_iphone_cook_services ── Topic
     │    │    │    │         └─■──apple_china_iphone_apples_quarter ── Topic
     │    │    │    └─revenue_billion_million_percent_netflix
     │    │    │         ├─revenue_percent_quarter_cloud_samsung
     │    │    │         │    ├─■──samsung_electronics_china_profit_productio
     │    │    │         │    └─cloud_revenue_percent_quarter_million
     │    │    │         │         ├─■──euros_percent_million_revenue_sap ──
     │    │    │         │         └─■──cloud_revenue_oracle_quarter_billion
     │    │    │         └─netflix_streaming_disney_tv_million
     │    │    │              ├─million_revenue_investors_twitter_company
     │    │    │              │    ├─■──slack_venture_capital_investors_start
     │    │    │              │    └─twitter_revenue_snap_advertising_alphabe
     │    │    │              │         ├─■──alphabet_yahoo_google_alphabets_
     │    │    │              │         └─■──twitter_snap_revenue_dorsey_mill
     │    │    │              └─■──netflix_streaming_disney_tv_fox ── Topic:
     │    │    └─amazon_amazons_prime_bezos_ecommerce
     │    │         ├─amazon_prime_amazons_ecommerce_sellers
     │    │         │    ├─■──flipkart_india_ecommerce_indian_amazon ── Topic
     │    │         │    └─amazon_prime_amazons_stores_foods
     │    │         │         ├─■──amazon_amazons_tickets_advertising_sellers
     │    │         │         └─■──amazon_prime_stores_foods_store ── Topic:
     │    │         └─amazon_bezos_city_amazons_seattle
     │    │              ├─■──amazon_fedex_delivery_shipping_ups ── Topic: 10
     │    │              └─amazon_bezos_city_seattle_amazons
     │    │                   ├─■──bezos_amazon_jeff_mackenzie_letter ── Topi
```

```
│    │                       └─■──amazon_city_seattle_headquarters_housing —
│    └─uber_huawei_chinese_company_bitcoin
│         ├─uber_drivers_ubers_lyft_kalanick
│         │    ├─uber_drivers_ubers_kalanick_city
│         │    │    ├─uber_kalanick_ubers_waymo_levandowski
│         │    │    │    ├─■──uber_waymo_levandowski_waymos_selfdriving
│         │    │    │    └─■──uber_kalanick_ubers_benchmark_board ── Top
│         │    │    └─uber_drivers_city_scooters_cities
│         │    │         ├─■──uber_scooters_city_scooter_transit ── Topi
│         │    │         └─uber_drivers_didi_ridehailing_taxi
│         │    │              ├─■──drivers_uber_taxi_london_lyft ── Topi
│         │    │              └─■──didi_ridehailing_uber_drivers_chuxing
│         │    └─softbank_lyft_wework_son_fund
│         │         ├─■──lyft_uber_lyfts_public_ipo ── Topic: 71
```

```
data_tech_sample['section'].unique()
```

```
array(['Technology News', 'Tech', 'Tech by VICE', 'technology',
       'Technology, Media and Telecommunications', 'tech', 'TECH',
       'Tech Guide', 'Tech Policy', 'Technology: Companies',
       'technology-quarterly', 'Technology Executive Council',
       'Tech Drivers'], dtype=object)
```

```
topics_per_class = model_tech.topics_per_class(docs_tech, classes=data_tech_sample
```

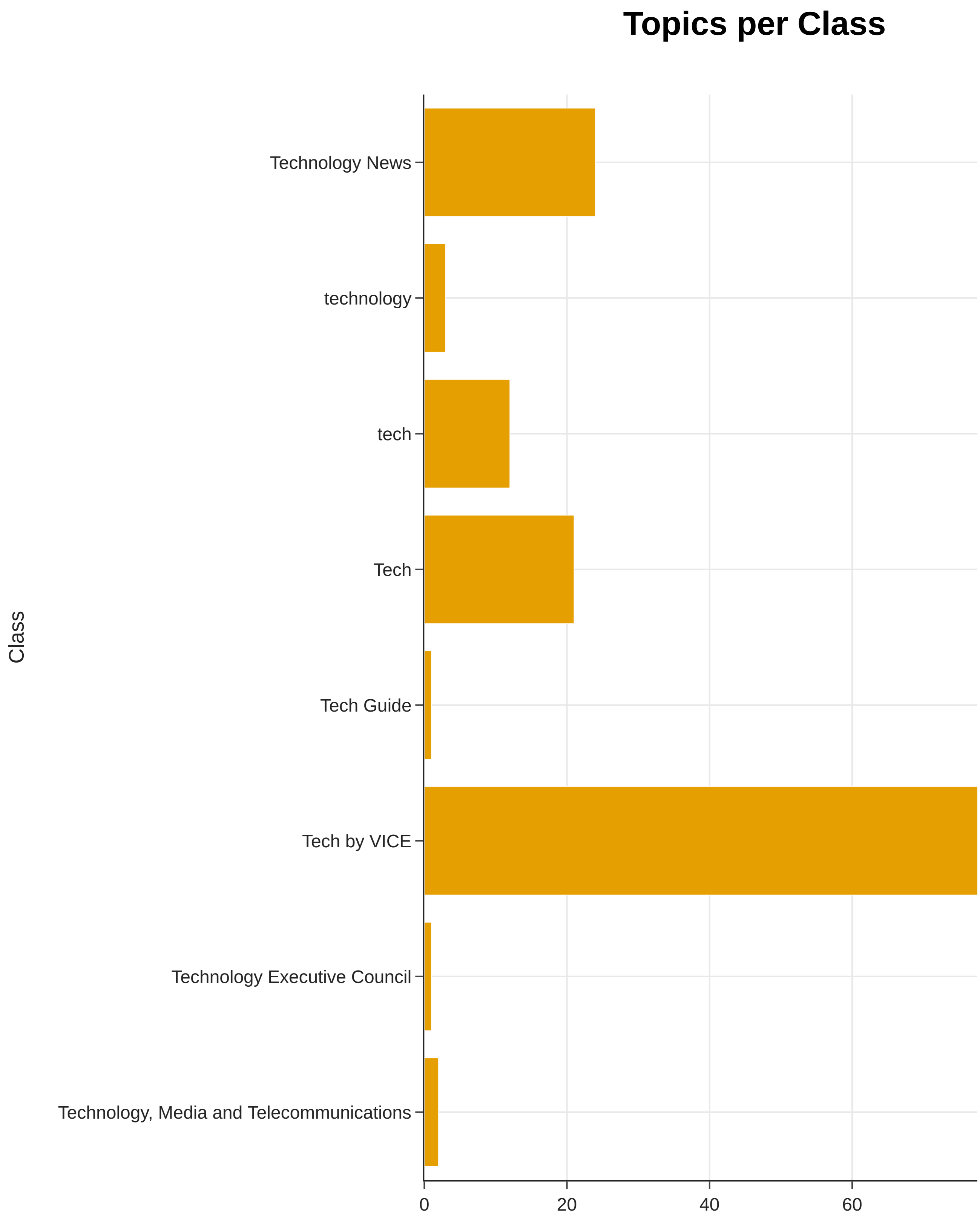
```
13it [00:04,  3.19it/s]
```

```
topics_per_class
```

	Topic	Words	Frequency	Class
0	-1	coverage, reuters, text, source, ltd	91	Technology, Media and Telecommunications
1	0	cup, fifa, telstar, fenech, ball	2	Technology, Media and Telecommunications
2	1	iss, stine, jacobsen, assange, clapper	2	Technology, Media and Telecommunications
3	5	bachelet, fantastic, vilified, toby, geneva	1	Technology, Media and Telecommunications
4	9	eikon, coverage, text, million, source	77	Technology, Media and Telecommunications
...	...	...	...	...
481	109	telefonica, vodafone, telenor, indra, operator	9	Technology News
482	-1	leeco, mcguire, leecos, property, counsel	1	Technology: Companies

property, counsel

```
model_tech.visualize_topics_per_class(topics_per_class, top_n_topics=20)
```



Frequency

Dynamic Modeling

- Analyzing the evolution of topics over time.
- How a topic is represented across different times.

```
#only extract month and year
# data_tech_sample['date'] = data_tech_sample['date'].dt.to_period('M')
timestamps = data_tech_sample['date'].to_list()
```

```
data_tech_sample['date']
```

```
0      2018-09-27 00:00:00
1      2017-05-18 00:00:00
2      2017-09-21 00:00:00
3      2019-07-02 12:00:00
4      2018-10-30 17:53:00
               ...
5995   2019-04-08 00:00:00
5996   2019-06-21 00:00:00
5997   2019-05-23 00:00:00
5998   2018-09-06 12:52:00
5999   2018-04-23 00:00:00
Name: date, Length: 6000, dtype: datetime64[ns]
```

```
len(data_tech_sample['date'].unique())
```

```
3209
```

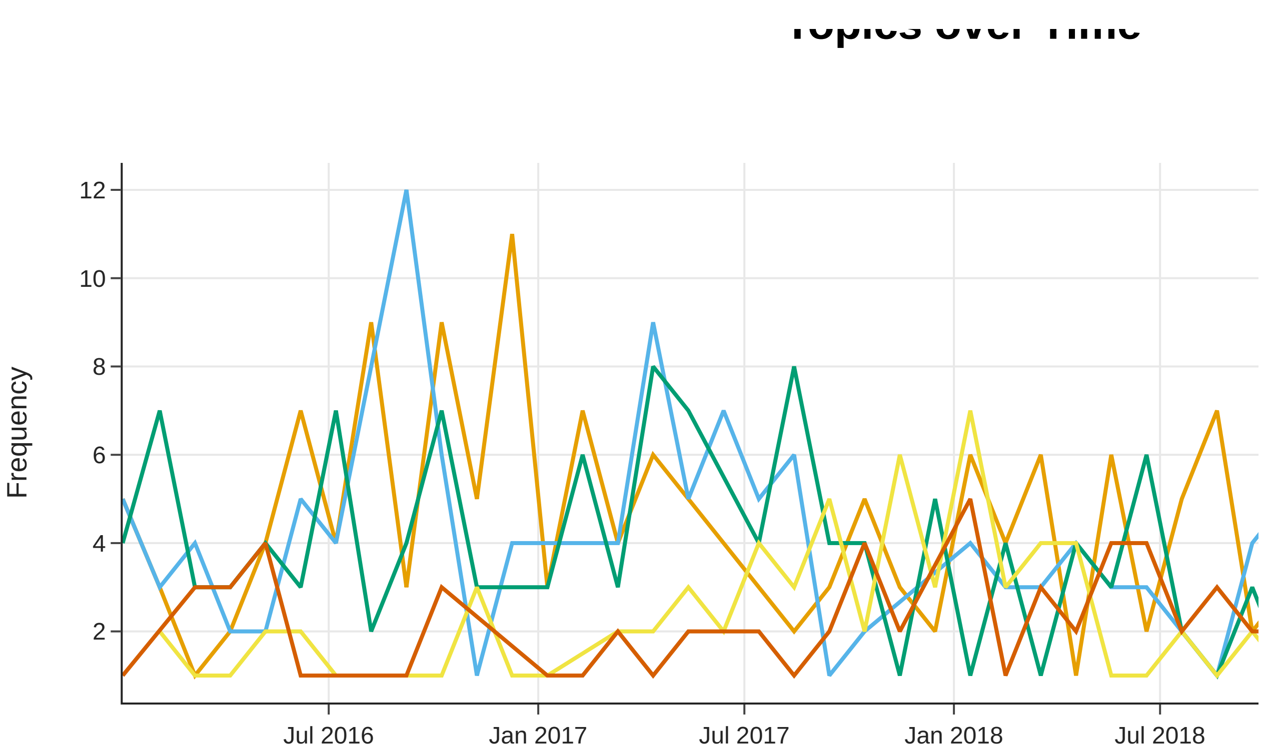
```
topics_over_time = model_tech.topics_over_time(docs_tech, timestamps, datetime_for
```

```
50it [00:29,  1.71it/s]
```

Visualize Topic Over time

```
model_tech.visualize_topics_over_time(topics_over_time, top_n_topics=5)
```

Topics over Time



Save Model

```
model_tech.save("GLG_V1_tech_topics_model")
```

Make Prediction

```
test_tech_data = data_tech.sample(n=10, random_state=3, ignore_index=True)
```

```
test_tech_data = test_tech_data['article'].to_list()
```

```
topics_tech, probs_tech = model_tech.transform(test_tech_data)
```

Batches: 100% 1/1 [00:00<00:00, 1.19it/s]

```
2022-09-29 16:18:38,496 - BERTopic - Reduced dimensionality
2022-09-29 16:18:38,502 - BERTopic - Predicted clusters
```

```
for i in topics_tech:
```

```
  print(model_tech.get_topic(i))
```

```
[('electric', 0.05555727348587223), ('cars', 0.0238663417942288), ('vehicles'
[('new', 0.005222939804304491), ('company', 0.005114758506380016), ('facebook
[('tax', 0.10266485021334065), ('irs', 0.030301573390024233), ('france', 0.02
[('new', 0.005222939804304491), ('company', 0.005114758506380016), ('facebook
[('facebook', 0.04448424639191789), ('zuckerberg', 0.04048538873566221), ('ca
[('electric', 0.05555727348587223), ('cars', 0.0238663417942288), ('vehicles'
[('brexit', 0.05047585241550907), ('fintech', 0.04993090890983098), ('uk', 0.
[('new', 0.005222939804304491), ('company', 0.005114758506380016), ('facebook
[('windows', 0.04392264763145568), ('apps', 0.02645388589873797), ('microsoft
[('bank', 0.04506502755552442), ('swift', 0.026600020727371426), ('credit', 0
```

```
model_tech.get_topic(3)
```

```
[('bitcoin', 0.0436503190776091),
 ('blockchain', 0.023942840528972182),
 ('cryptocurrency', 0.023371896132604233),
 ('libra', 0.02162769926467183),
 ('banks', 0.01578934939551688),
 ('currency', 0.014700107155199543),
 ('digital', 0.014682047403972257),
 ('cryptocurrencies', 0.012647020376028393),
 ('ethereum', 0.012311332403242628),
 ('financial', 0.011613045709468651)]
```

Cluster Health Article Sample Data and Generate Topics

- Load Health Article Sample Data
- Apply the all-mpnet-base-v2 model which provides the best quality for sentence embedding.
- Cluster document using HDBSCAN clustering algorithm
- Apply C-TF-IDF on each cluster and generate Topics for each cluster
- Dynamic Topics Over period of time

```
# load 6000 Health Article sample data
data_health = data_tech_health[data_tech_health['tech_health_tag']=='health']
data_health_sample = data_health.sample(n=6000, random_state=42, ignore_index=True
```

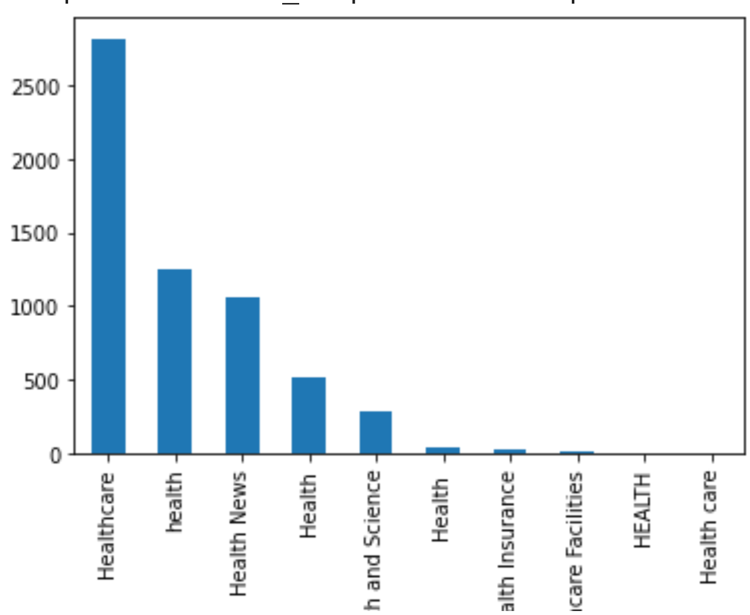
```
data_health_sample.head()
```

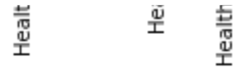
	date	year	month	day	title	article
						reuters health

0	2019-11-21 00:00:00	2019	11.0	21	diets with more fiber yogurt tied to lower risk of lung cancer	– even among smokers people eat fiber yogurt may less ly develop lung cancer don't consume much foods research review suggests. res...	https://www.reuters.c /article/us-hea lungcancer-yogu fiber/diets-with-mo fiber-yogurt-tied- lower-risk-of-lu canc idUSKBN1XV2
1	2020-03-02 00:00:00	2020	3.0	2	briefwestleaf officially rebrands as decibel cannabis company	march 2 reuters westleaf inc westleaf inc. officially rebrands decibel cannabis company inc. source text eikon company coverage	https://www.reuters.c /article/brief-westle officially-rebrands- de/brief-westle officially-rebrands- decibel-cannal compa idUSFWN2A\
2	2019-03-04 00:00:00	2019	3.0	4	mmr vaccine does not cause autism another study	cnnthe measles mumps rubella vaccine increase risk autism trigger autism children risk according new study	https://www.cnn.c /2019/03/04/hea /mmr-vaccine-autis study/index.h

```
data_health_sample['section'].value_counts().plot.bar()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f653844f050>





Train BERTopic Model

```
# Train our topic model using our pre-trained sentence-transformers embeddings
# nr_topics="auto" merge similar topics

model_health = BERTopic(verbose=True)

#convert to list
docs_health = data_health_sample.article.to_list()

topics_health, probabilities = model_health.fit_transform(docs_health)
```

Batches: 100% 188/188 [04:49<00:00, 3.60it/s]

```
2022-09-29 17:16:44,973 - BERTopic - Transformed documents to Embeddings
2022-09-29 17:16:57,900 - BERTopic - Reduced dimensionality
2022-09-29 17:16:58,178 - BERTopic - Clustered reduced embeddings
```

Select Top Topics

```
# After training the model, you can access the size of topics in descending order
model_health.get_topic_freq()
```

	Topic	Count
0	-1	1786
1	0	316
2	1	276
3	2	205
4	3	161
...	...	...
99	98	12
100	99	11
101	100	11
102	101	11
103	102	11

104 rows × 2 columns

Note:

Topic -1 is the largest and it refers to outliers tweets that do not assign to any topics generated. In this case, we will ignore Topic -1.

Select One Topic

- You can select a specific topic and get the top n words for that topic and their c-TF-IDF scores

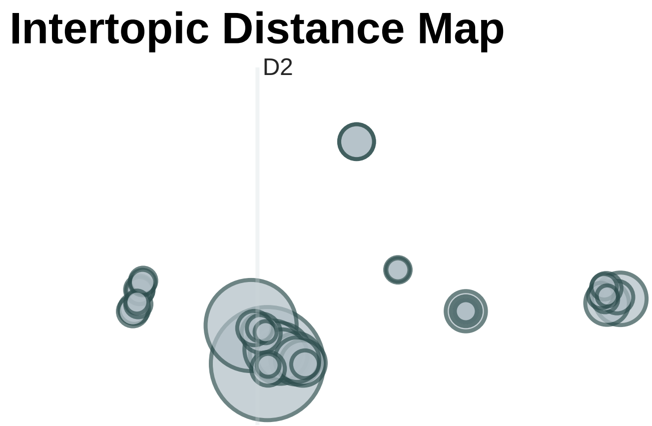
```
model_health.get_topic(0)
```

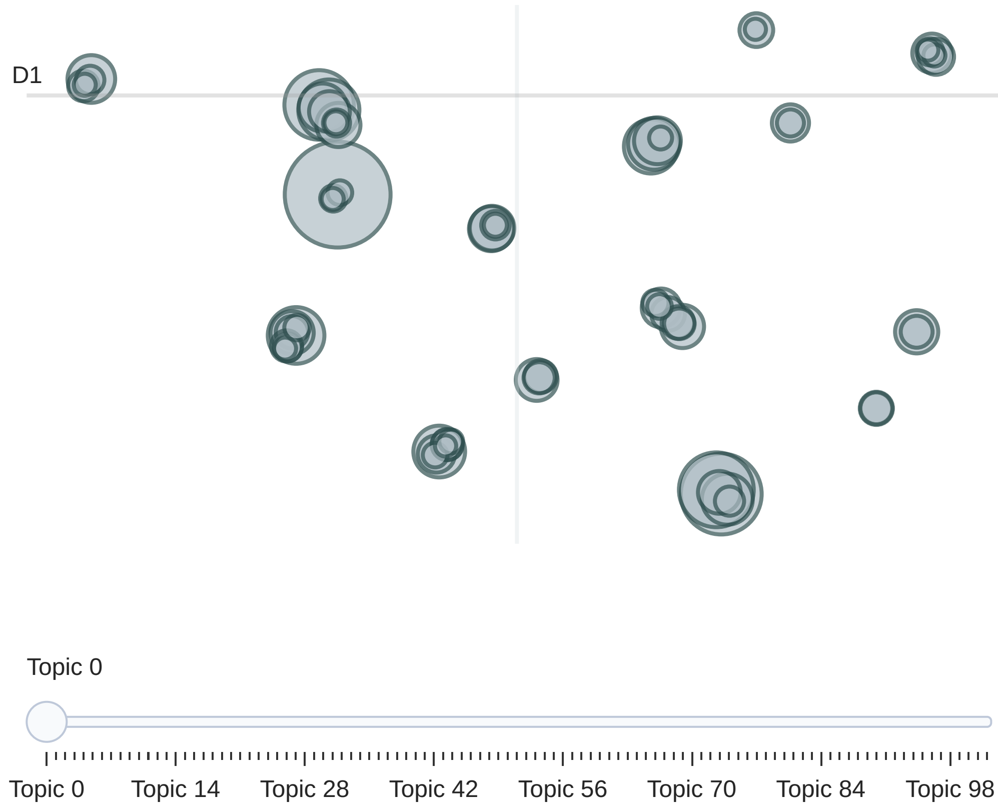
```
[('march', 0.04144458097026481),
 ('covid19', 0.03435432177587527),
 ('eikon', 0.03132331992520448),
 ('2020', 0.030673188923154443),
 ('impact', 0.02963488566970456),
 ('text', 0.029630598481679656),
 ('source', 0.028561437109156353),
 ('coverage', 0.02808645417670462),
 ('company', 0.027528623022715856),
 ('reuters', 0.025783371484713803)]
```

Visualize Topics

Visualize topics generated with their sizes and corresponding words

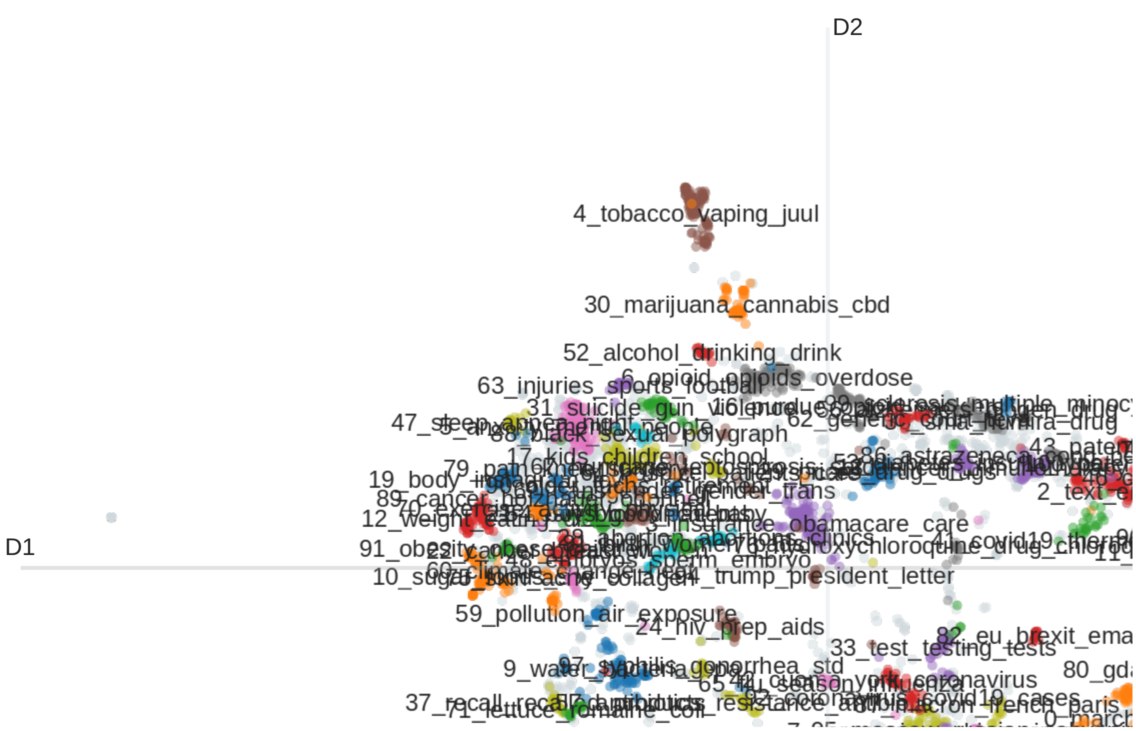
```
model_health.visualize_topics()
```



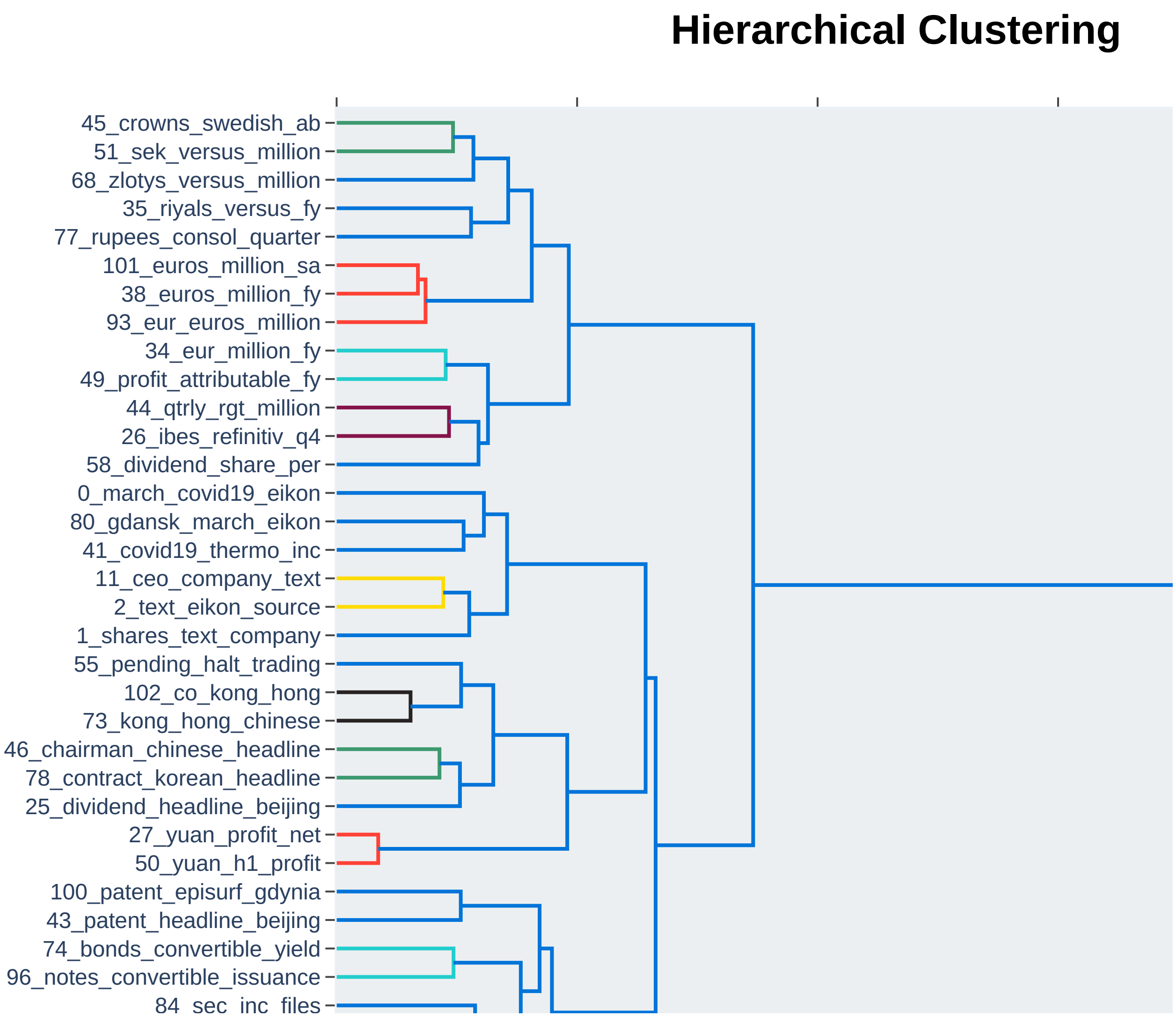


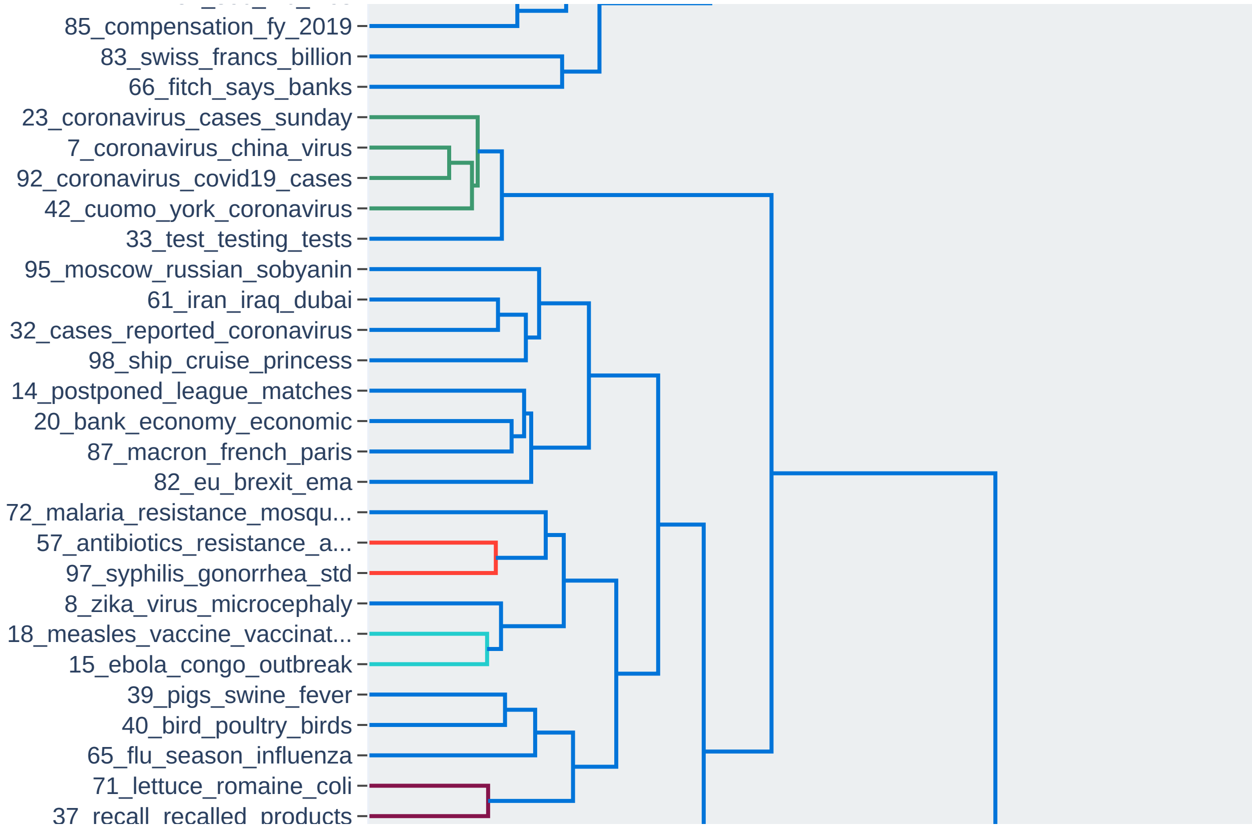

```
model_health.visualize_documents(docs_health)
```

Documents and




```
model_health.visualize_hierarchy()
```

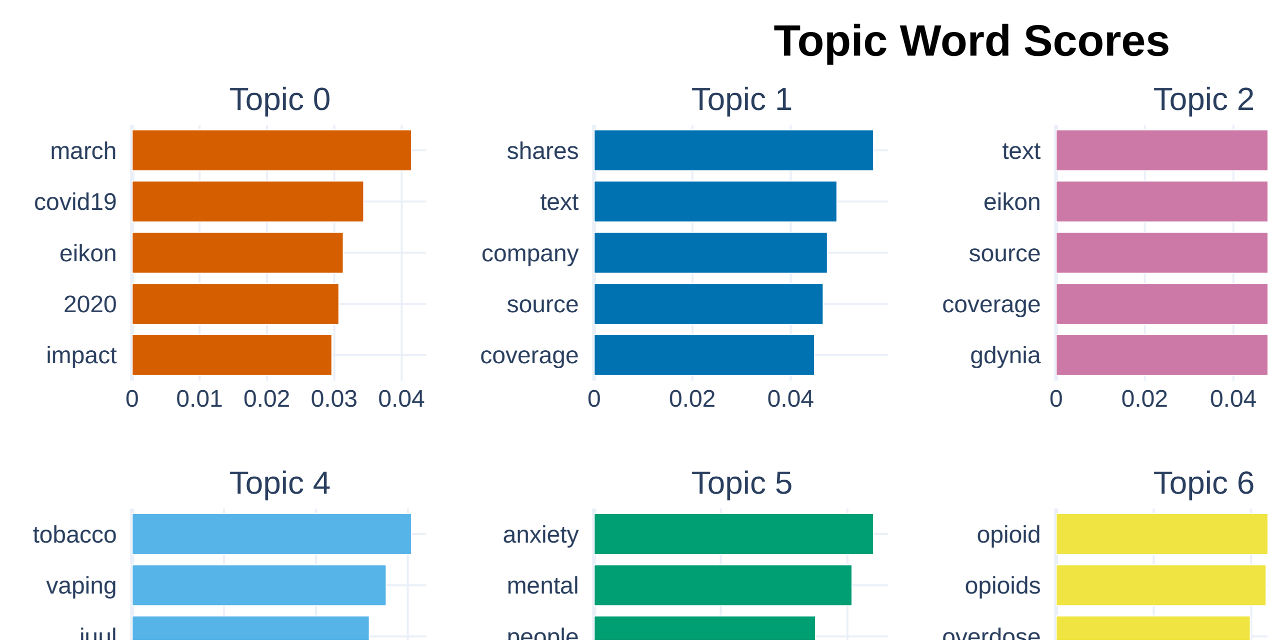


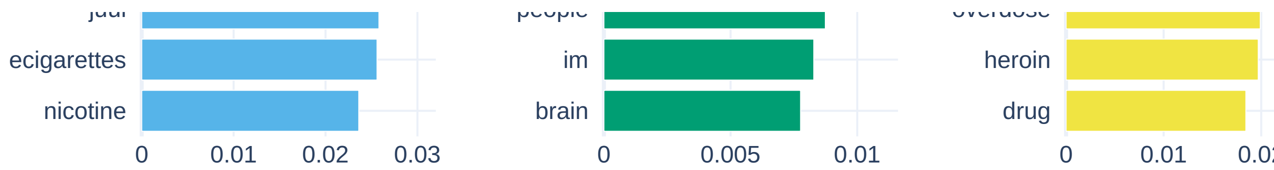


Visualize Terms

The visualize_barchart method will show the selected terms for a few topics by creating bar charts out of the c-TF-IDF scores.

```
model_health.visualize_barchart()
```





Visualize Topic Similarity

```
model_health.visualize_heatmap()
```

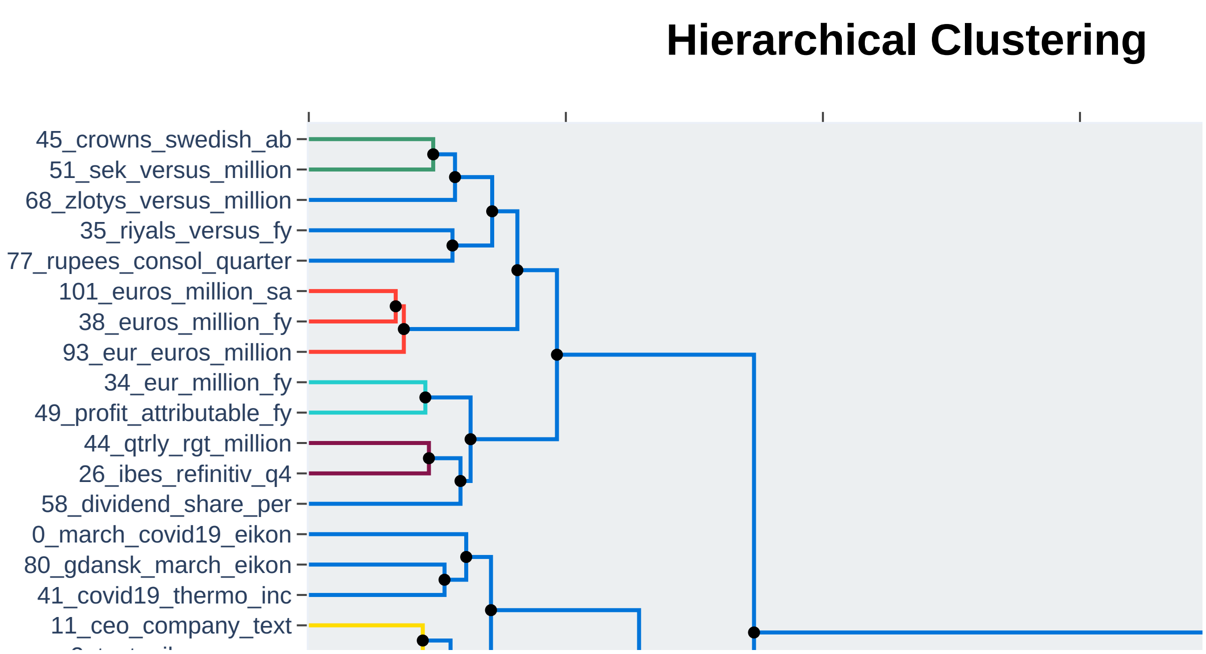
Hirarical Topics

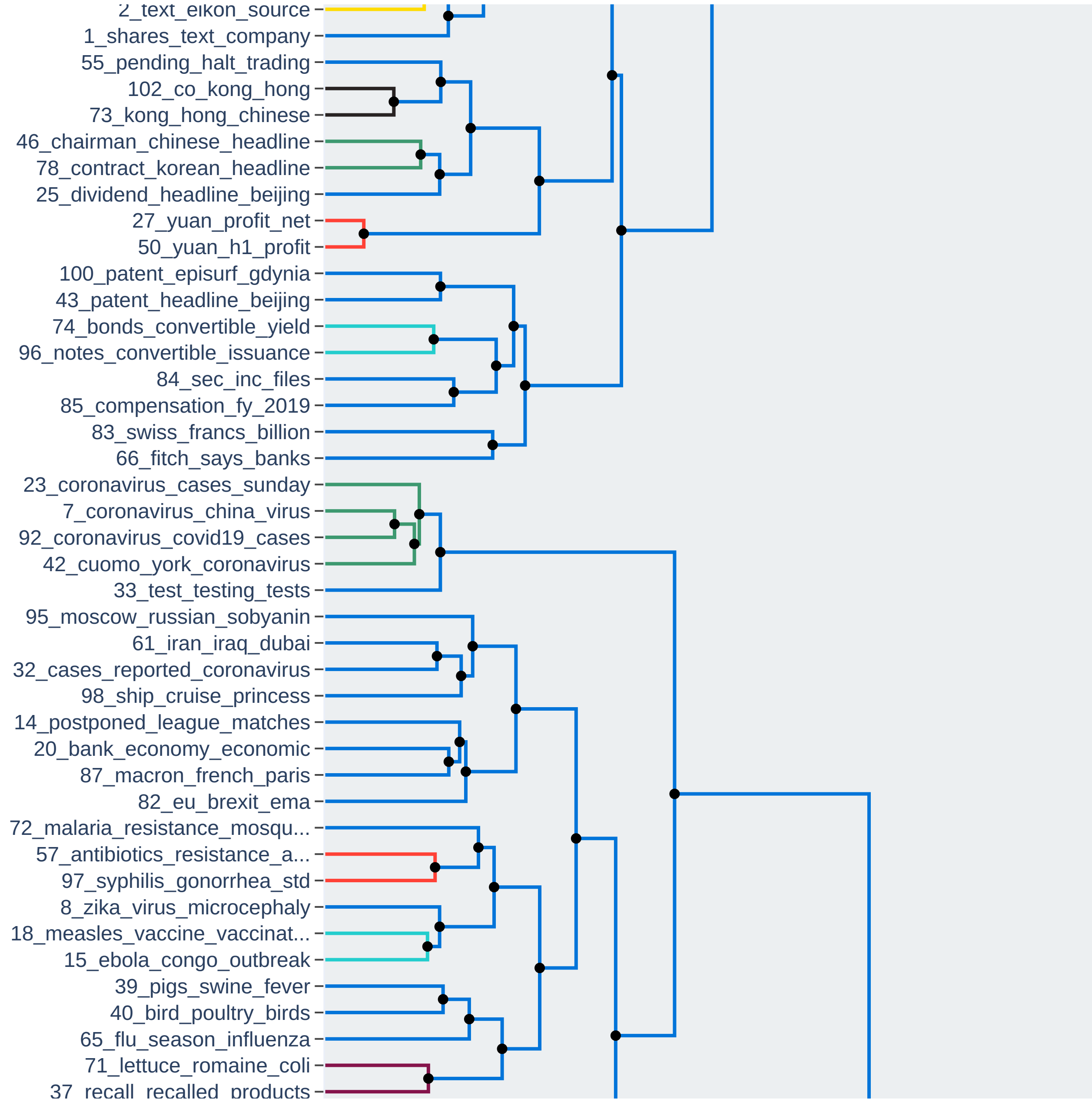
```
hierarchical_topics = model_health.hierarchical_topics(docs_health)
```

```
100%|██████████| 102/102 [00:01<00:00, 97.39it/s]
```

```
hierarchical_topics
```

```
model_health.visualize_hierarchy(hierarchical_topics=hierarchical_topics)
```






```
tree = model_health.get_topic_tree(hierarchical_topics)
print(tree)
```

```
.
├─text_source_coverage_company_million
│    ├─million_versus_ago_fy_net
│    │    ├─million_versus_ago_euros_crowns
│    │    │    ├─versus_million_ago_crowns_rupees
│    │    │    │    ├─crowns_zlotys_sek_million_versus
│    │    │    │    │    ├─crowns_sek_ab_million_swedish
│    │    │    │    │    │    ├─■──crowns_swedish_ab_million_gdynia ── Topic:
│    │    │    │    │    │    └─■──sek_versus_million_loss_ab ── Topic: 51
│    │    │    │    │    └─■──zlotys_versus_million_ago_sa ── Topic: 68
│    │    │    │    └─rupees_versus_profit_million_riyals
│    │    │    │         ├─■──riyals_versus_fy_million_egp ── Topic: 35
│    │    │    │         └─■──rupees_consol_quarter_profit_billion ── Topic:
│    │    │    └─euros_million_versus_eur_fy
│    │    │         └─euros_million_versus_fy_ago
```

```
│    │    │         ├─euros_million_versus_fy_ago
│    │    │         │    ├─■─euros_million_sa_gdynia_revenue ── Topic: 101
│    │    │         │    └─■─euros_million_fy_versus_loss ── Topic: 38
│    │    │         └─■─eur_euros_million_ebitda_dkk ── Topic: 93
│    │    └─million_2020_eur_2019_eikon
│    │         ├─eur_million_fy_profit_2020
│    │         │    ├─■─eur_million_fy_2020_2019 ── Topic: 34
│    │         │    └─■─profit_attributable_fy_ltd_eikon ── Topic: 49
│    │         └─qtrly_q4_ibes_refinitiv_million
│    │              ├─qtrly_q4_ibes_refinitiv_million
│    │              │    ├─■─qtrly_rgt_million_insr_revenue ── Topic: 44
│    │              │    └─■─ibes_refinitiv_q4_2019_earnings ── Topic: 26
│    │              └─■─dividend_share_per_proposes_board ── Topic: 58
│    └─text_source_coverage_company_reuters
│         ├─text_source_coverage_company_reuters
│         │    ├─text_source_coverage_company_eikon
│         │    │    ├─march_covid19_eikon_text_2020
│         │    │    │    ├─■─march_covid19_eikon_2020_impact ── Topic: 0
│         │    │    │    └─covid19_march_eikon_text_source
│         │    │    │         ├─■─gdansk_march_eikon_text_agm ── Topic: 80
│         │    │    │         └─■─covid19_thermo_inc_sarscov2_eikon ── Topic
│         │    │    └─text_source_coverage_company_eikon
│         │    │         ├─text_eikon_source_coverage_company
│         │    │         │    ├─■─ceo_company_text_eikon_source ── Topic: 11
│         │    │         │    └─■─text_eikon_source_coverage_gdynia ── Topic
│         │    │         └─■─shares_text_company_source_coverage ── Topic: 1
│         │    └─yuan_chinese_co_ltd_headline
│         │         ├─ltd_co_dividend_chinese_text
│         │         │    ├─hong_kong_ltd_trading_co
│         │         │    │    ├─■─pending_halt_trading_announcement_ltd ── T
│         │         │    │    └─kong_hong_co_chinese_newsroom
│         │         │    │         ├─■─co_kong_hong_resume_ltd ── Topic: 102
│         │         │    │         └─■─kong_hong_chinese_newsroom_co ── Topi
│         │         │    └─dividend_headline_beijing_co_ltd
│         │         │         ├─headline_co_beijing_chinese_ltd
│         │         │         │    ├─■─chairman_chinese_headline_co_resigned
│         │         │         │    └─■─contract_korean_headline_co_ltd ── To
│         │         │         └─■─dividend_headline_beijing_shares_sharehold
│         │         └─yuan_profit_net_million_chinese
│         │              ├─■─yuan_profit_net_million_chinese ── Topic: 27
│         │              └─■─yuan_h1_profit_net_million ── Topic: 50
│         └─fitch_patent_text_source_says
│              ├─patent_text_source_convertible_inc
│              │    ├─patent_headline_beijing_korean_text
```

```
data_health_sample['section'].unique()
```

```
array(['Health News', 'Healthcare', 'health', 'Health',
       'Health Insurance', 'Health and Science', 'Health ',
       'Healthcare Facilities', 'HEALTH', 'Health care'], dtype=object)
```

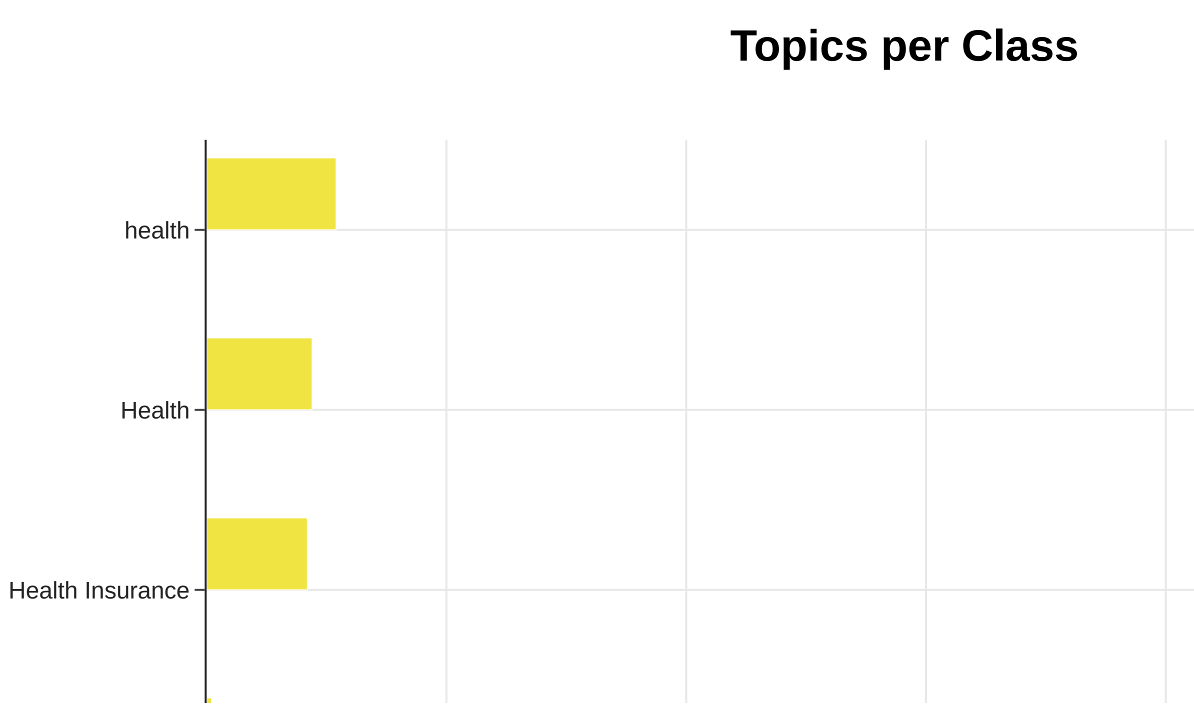
```
topics_per_class = model_health.topics_per_class(docs_health, classes=data_health_
```

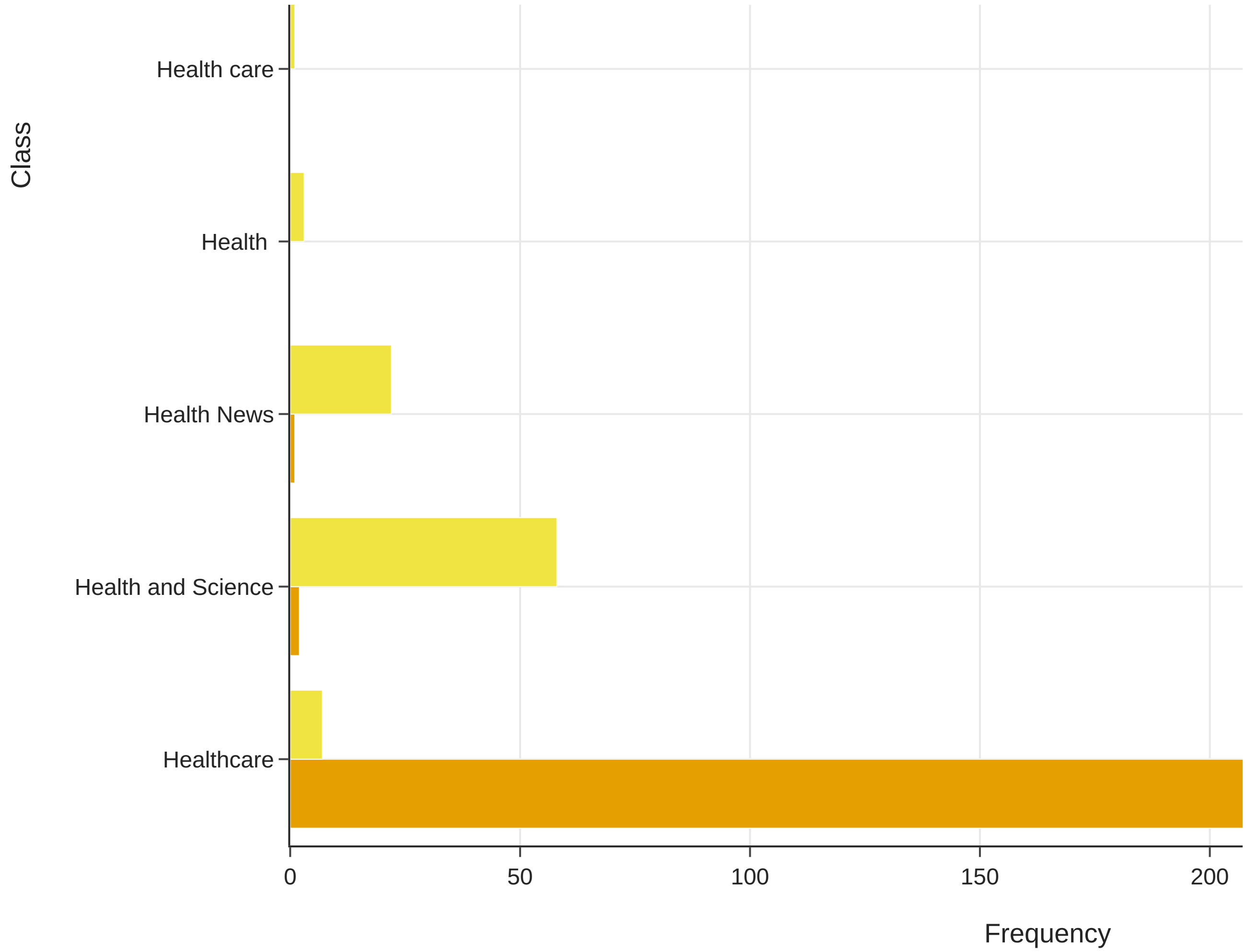
```
10it [00:03,  3.32it/s]
```

```
topics_per_class
```

	Topic	Words	Frequency	Class
0	-1	psychedelic, psychedelics, drag, placebo, people	13	Health
1	3	healthcare, millennials, care, insurance, system	3	Health
2	4	vape, vaping, cigarettes, tobacco, nicotine	4	Health
3	5	anime, selfcare, im, way, work	4	Health
4	7	sarscov2, virus, cleavage, garry, furin	1	Health
...	...	...	...	...
312	92	pregnant, coronavirus, washington, virus, travel	5	Health News
313	95	moscow, sobyanin, russian, moscows, coronavirus	1	Health News
314	97	hpv, syphilis, gonorrhoea, sexually, infections	2	Health News
315	98	cruise, ship, kolstoe, princess, crew	2	Health News

```
model_health.visualize_topics_per_class(topics_per_class, top_n_topics=20)
```





Dynamic Modeling

- Analyzing the evolution of topics over time.
- How a topic is represented across different times.

```
#only extract month and year
timestamps = data_health_sample['date'].to_list()
```

```
data_health_sample['date']
```

```
0       2019-11-21 00:00:00
1       2020-03-02 00:00:00
2       2019-03-04 00:00:00
3       2017-05-19 00:00:00
4       2016-08-11 20:38:00
                ...
5995    2018-02-02 00:00:00
5996    2018-10-03 00:00:00
```

```
5997   2020-02-04 00:00:00
5998   2017-05-12 00:00:00
5999   2017-12-14 00:00:00
Name: date, Length: 6000, dtype: datetime64[ns]
```

```
len(data_health_sample['date'].unique())
```

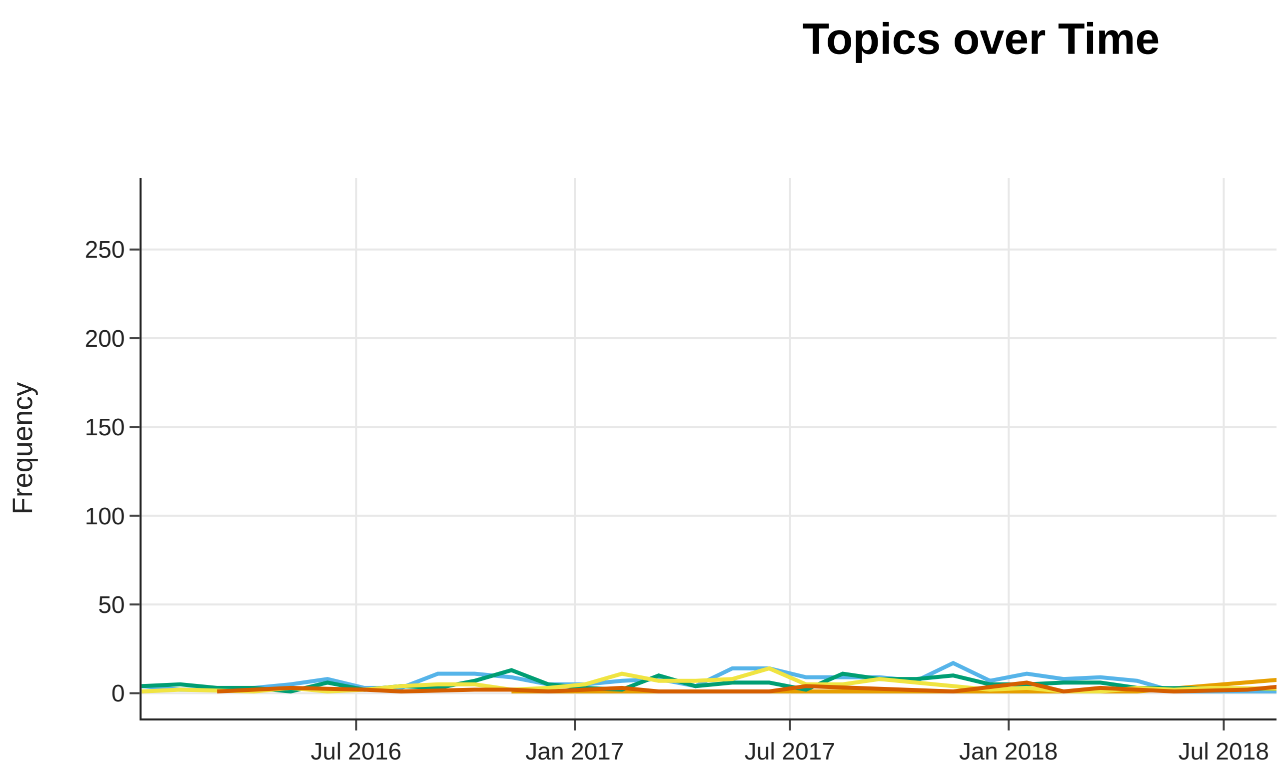
```
2005
```

```
topics_over_time = model_health.topics_over_time(docs_health, timestamps, datetime_
```

```
50it [00:20,  2.40it/s]
```

Visualize Topic Over time

```
model_health.visualize_topics_over_time(topics_over_time, top_n_topics=5)
```



Save Model

```
model_health.save("GLG_V1_health_topics_model")
```

Make Prediction

```
test_health_data = data_health.sample(n=10, random_state=3, ignore_index=True)
```

```
test_health_data = test_health_data['article'].to_list()
```

```
topics_test, probs_test = model_health.transform(test_health_data)
```

Batches: 100% 1/1 [00:00<00:00, 1.17it/s]

```
2022-09-29 17:22:44,134 - BERTopic - Reduced dimensionality
2022-09-29 17:22:44,138 - BERTopic - Predicted clusters
```

```
for i in topics_test:
  print(model_health.get_topic(i))
```

```
[('people', 0.005194395807170161), ('health', 0.004872328496367309), ('one',
[('body', 0.023360679745947188), ('instagram', 0.018949502739342185), ('love'
[('people', 0.005194395807170161), ('health', 0.004872328496367309), ('one',
[('bird', 0.07514759092480176), ('poultry', 0.06740052560381464), ('birds', 0
[('people', 0.005194395807170161), ('health', 0.004872328496367309), ('one',
[('people', 0.005194395807170161), ('health', 0.004872328496367309), ('one',
[('people', 0.005194395807170161), ('health', 0.004872328496367309), ('one',
[('body', 0.023360679745947188), ('instagram', 0.018949502739342185), ('love'
[('people', 0.005194395807170161), ('health', 0.004872328496367309), ('one',
[('heart', 0.045517921310255), ('blood', 0.020286131757340235), ('patients',
```

```
model_health.get_topic(3)
```

```
[('insurance', 0.02349201293285935),
 ('obamacare', 0.021828409850689016),
 ('care', 0.016554278758772068),
 ('bill', 0.01589964099961063),
 ('health', 0.014515894076677848),
 ('plans', 0.014051257004694694),
 ('plan', 0.012773641226400841),
 ('medicaid', 0.012530089582404065),
 ('would', 0.012522795571726584),
 ('coverage', 0.011097271823089344)]
```

Cluster Health and Technology Article Sample Data and Generate Topics

- Load Health and Technology Article Sample Data
- Apply the all-mpnet-base-v2 model which provides the best quality for sentence embedding.
- Cluster document using HDBSCAN clustering algorithm
- Apply C-TF-IDF on each cluster and generate Topics for each cluster
- Dynamic Topics Over period of time

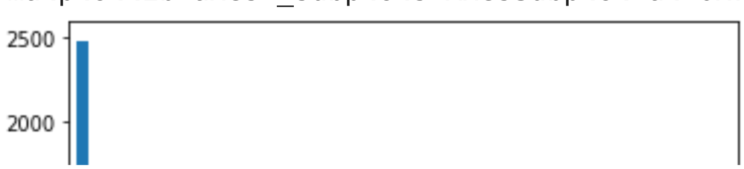
```
# load 6000 Health Article sample data
data_health_tech_sample = data_tech_health.sample(n=10000, random_state=42, ignore_
```

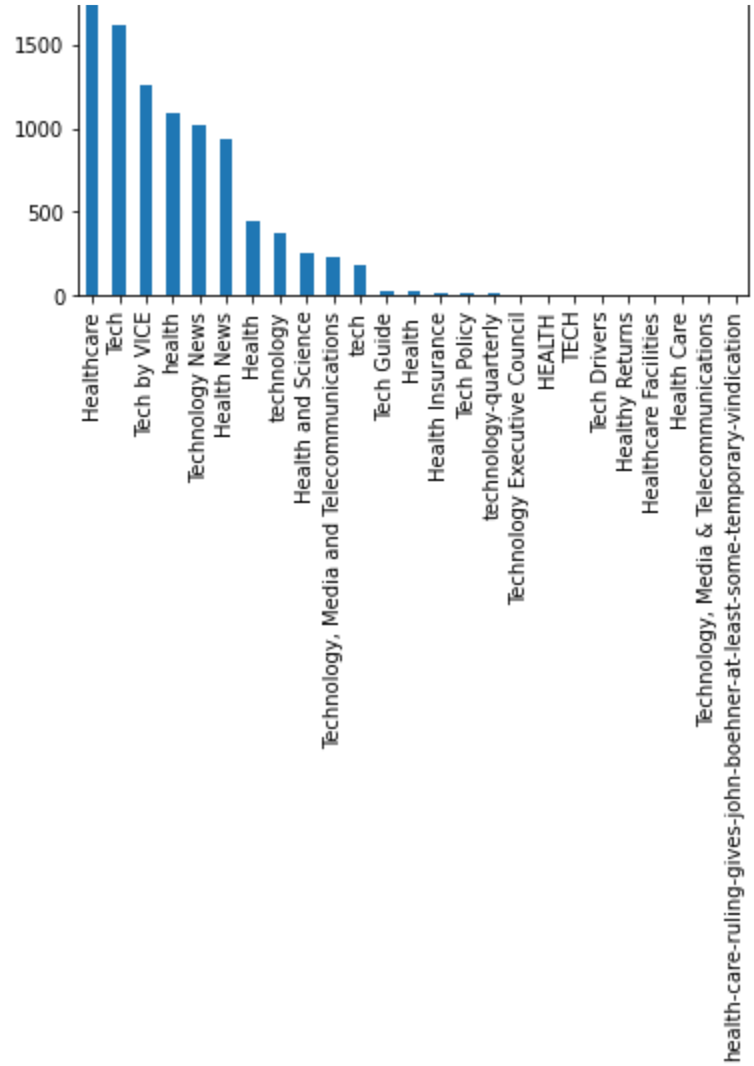
```
data_health_tech_sample.head(3)
```

	date	year	month	day	title	article	
0	2016-11-15 00:00:00	2016	11.0	15	nokia says telecoms gear market tough but outpacing ericsson	barcelonahelsinki reuters nokia nokia.he tuesday outperforming archrival ericsson ericb.st weak telecoms equipment market shares fell dividend pla...	http://www.r /article/nol idUSL
1	2019-07-23 15:33:55	2019	7.0	23	your data were ‘anonymized’? these scientists can still identify you	computer scientists developed algorithm pick almost american databases supposedly stripped personal information. medical records	https://www.n /2019/07/23/h privacy-prot

```
data_health_tech_sample['section'].value_counts().plot.bar()
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f6540697190>





Train BERTopic Model

```
# Train our topic model using our pre-trained sentence-transformers embeddings
# nr_topics="auto" merge similar topics

model_health_tech = BERTopic(verbose=True)

#convert to list
docs_health_tech = data_health_tech_sample.article.to_list()

topics_health_tech, probabilities = model_health_tech.fit_transform(docs_health_te
```

Batches: 100% 313/313 [09:17<00:00, 3.56it/s]

```
2022-09-29 16:53:08,083 - BERTopic - Transformed documents to Embeddings
2022-09-29 16:53:29,134 - BERTopic - Reduced dimensionality
2022-09-29 16:53:29,642 - BERTopic - Clustered reduced embeddings
```

Select Top Topics

```
# After training the model, you can access the size of topics in descending order
model_health_tech.get_topic_freq()
```

	Topic	Count
0	-1	3575
1	0	302
2	1	193
3	2	177
4	3	134
...	...	...
176	175	11
177	176	11
178	177	11
179	178	11
180	179	11

181 rows × 2 columns

Note:

Topic -1 is the largest and it refers to outliers tweets that do not assign to any topics generated. In this case, we will ignore Topic -1.

Select One Topic

- You can select a specific topic and get the top n words for that topic and their c-TF-IDF scores

```
model_health_tech.get_topic(0)
```

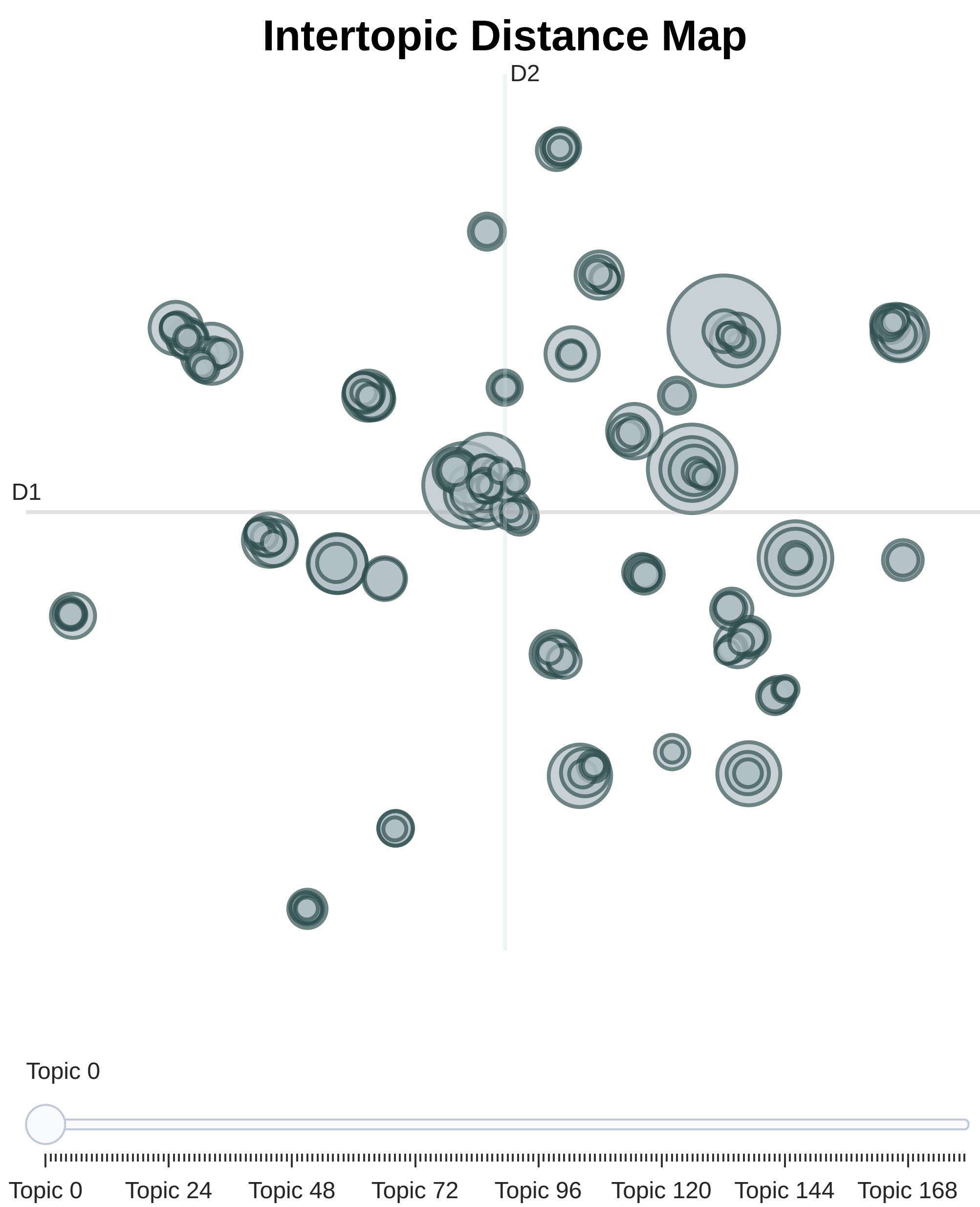
```
[('march', 0.04272153575069582),
 ('eikon', 0.038643526867302706),
 ('covid19', 0.03714104985038141),
 ('text', 0.03452203491744789),
 ('coverage', 0.034136303996588124),
```

```
('source', 0.03245557357868192),
('2020', 0.02761948611104587),
('impact', 0.02747982993237315),
('reuters', 0.02529173909159006),
('gdansk', 0.024745514089629835)]
```

Visualize Topics

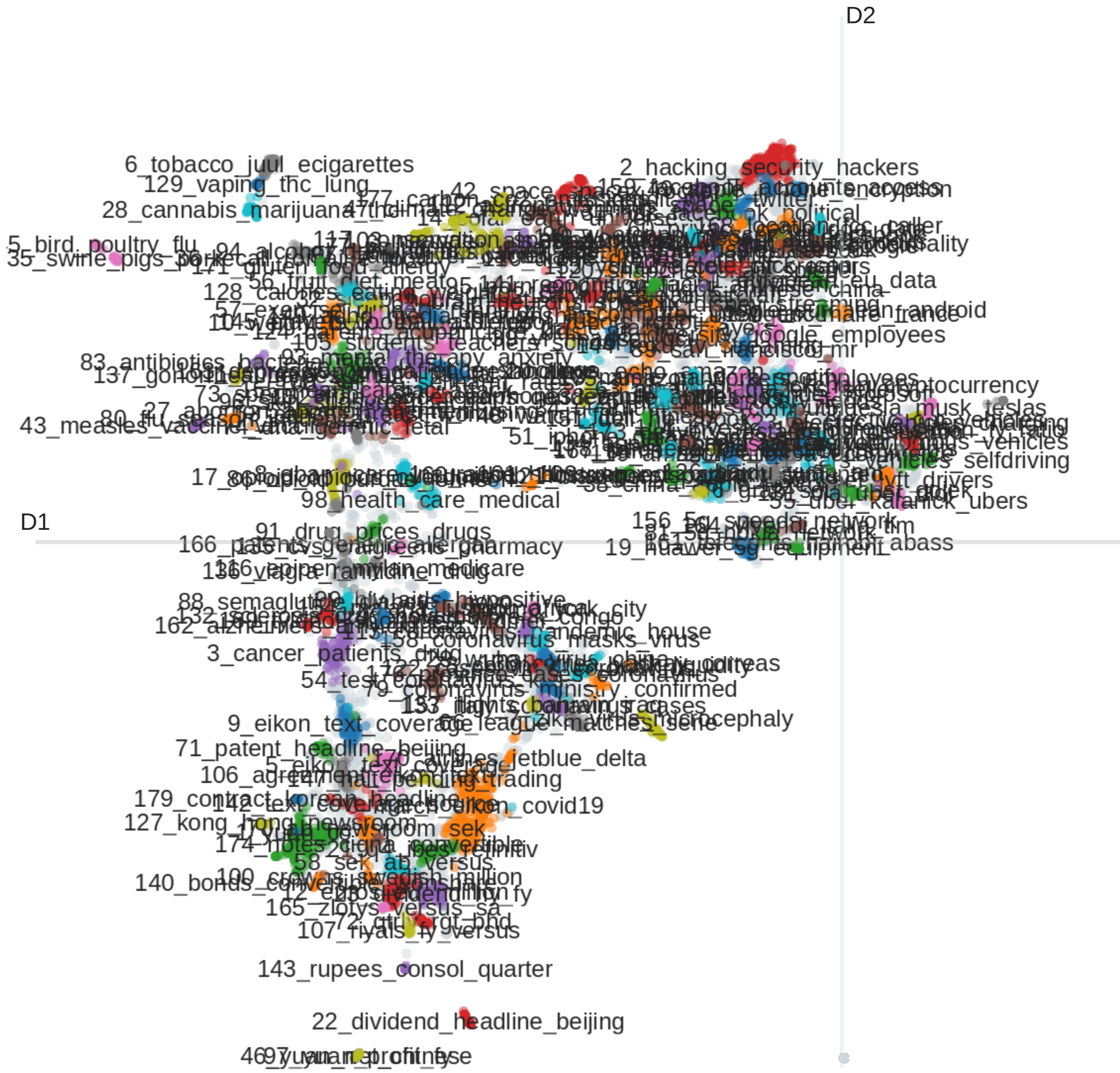
Visualize topics generated with their sizes and corresponding words

```
model_health_tech.visualize_topics()
```



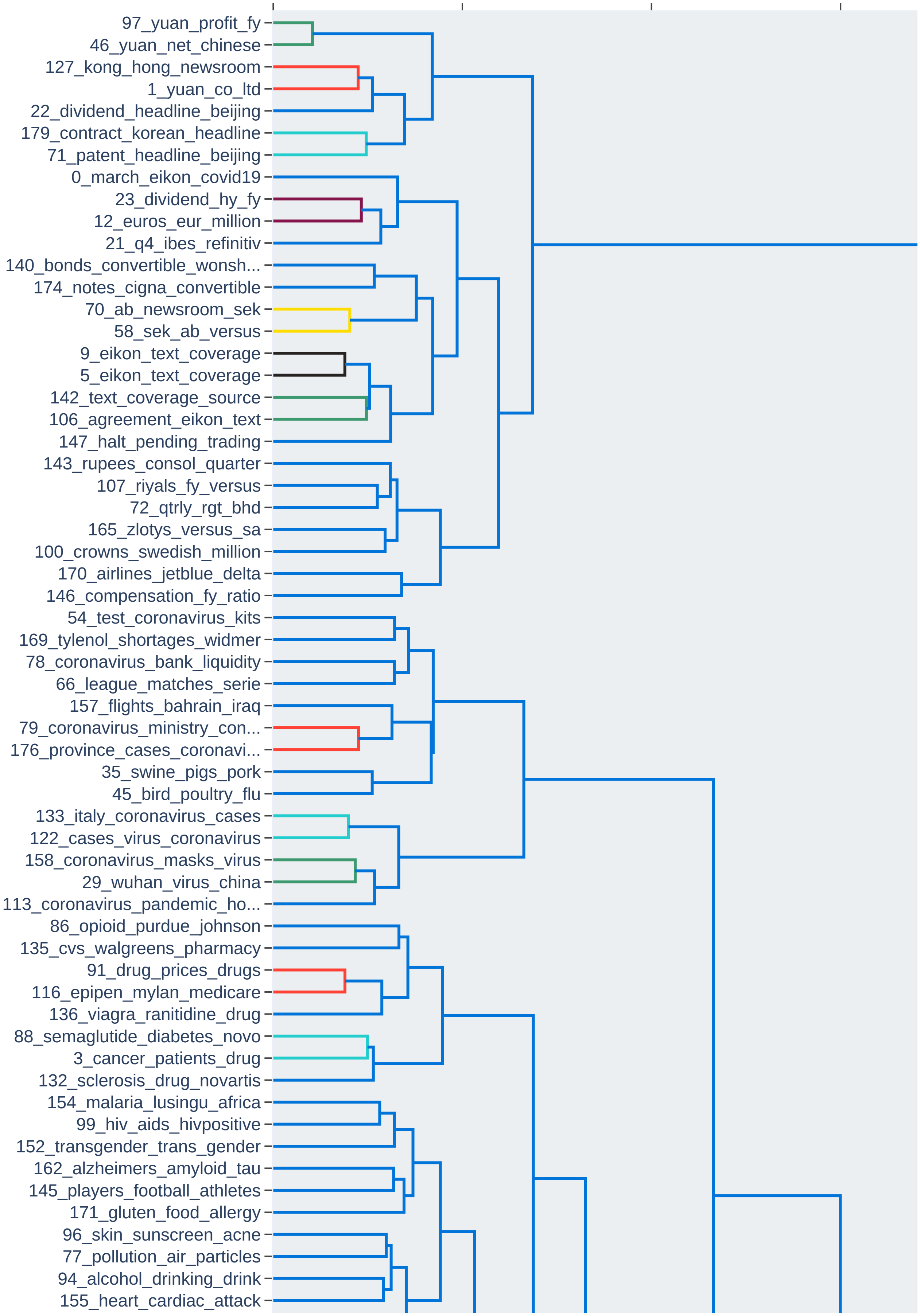

```
model_health_tech.visualize_documents(docs_health_tech)
```

Documents and




```
model_health_tech.visualize_hierarchy()
```

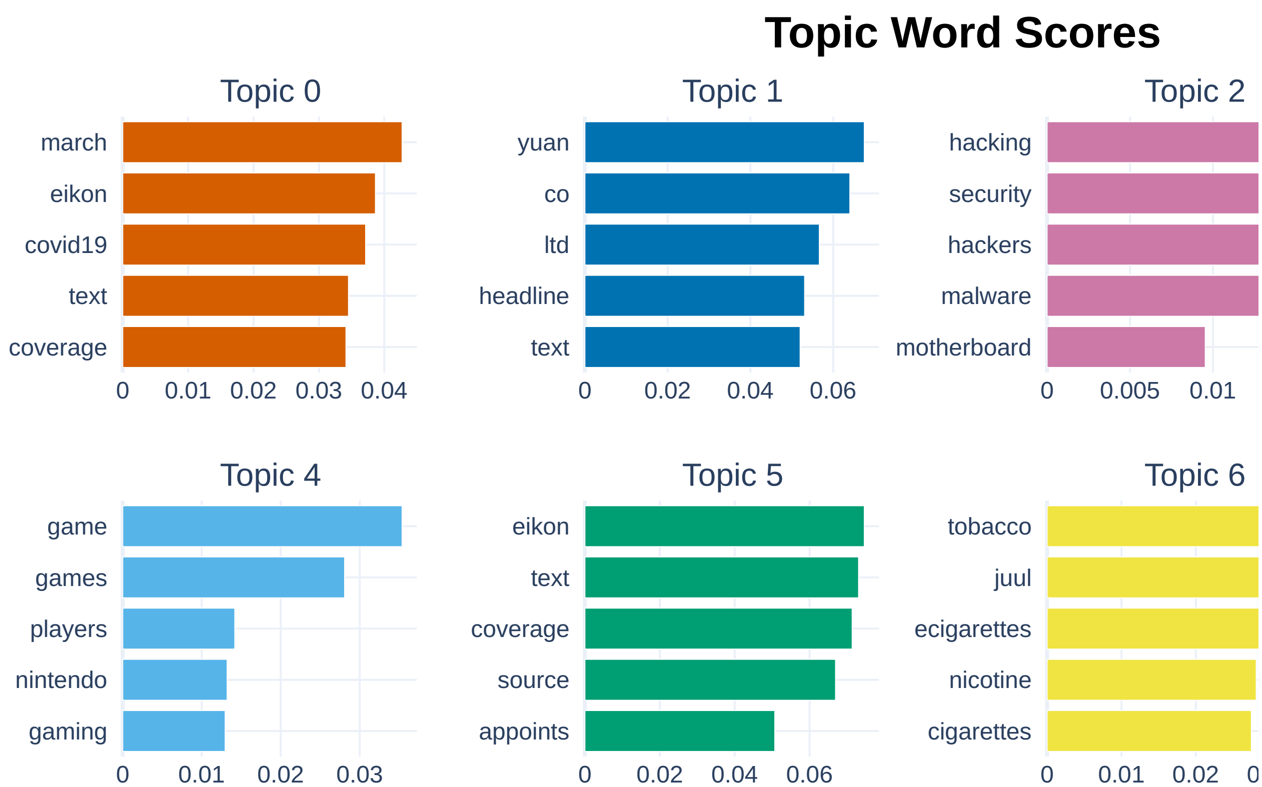
Hierarchical Clustering



Visualize Terms

The visualize_barchart method will show the selected terms for a few topics by creating bar charts out of the c-TF-IDF scores.

```
model_health_tech.visualize_barchart()
```



Visualize Topic Similarity

```
model_health_tech.visualize_heatmap()
```

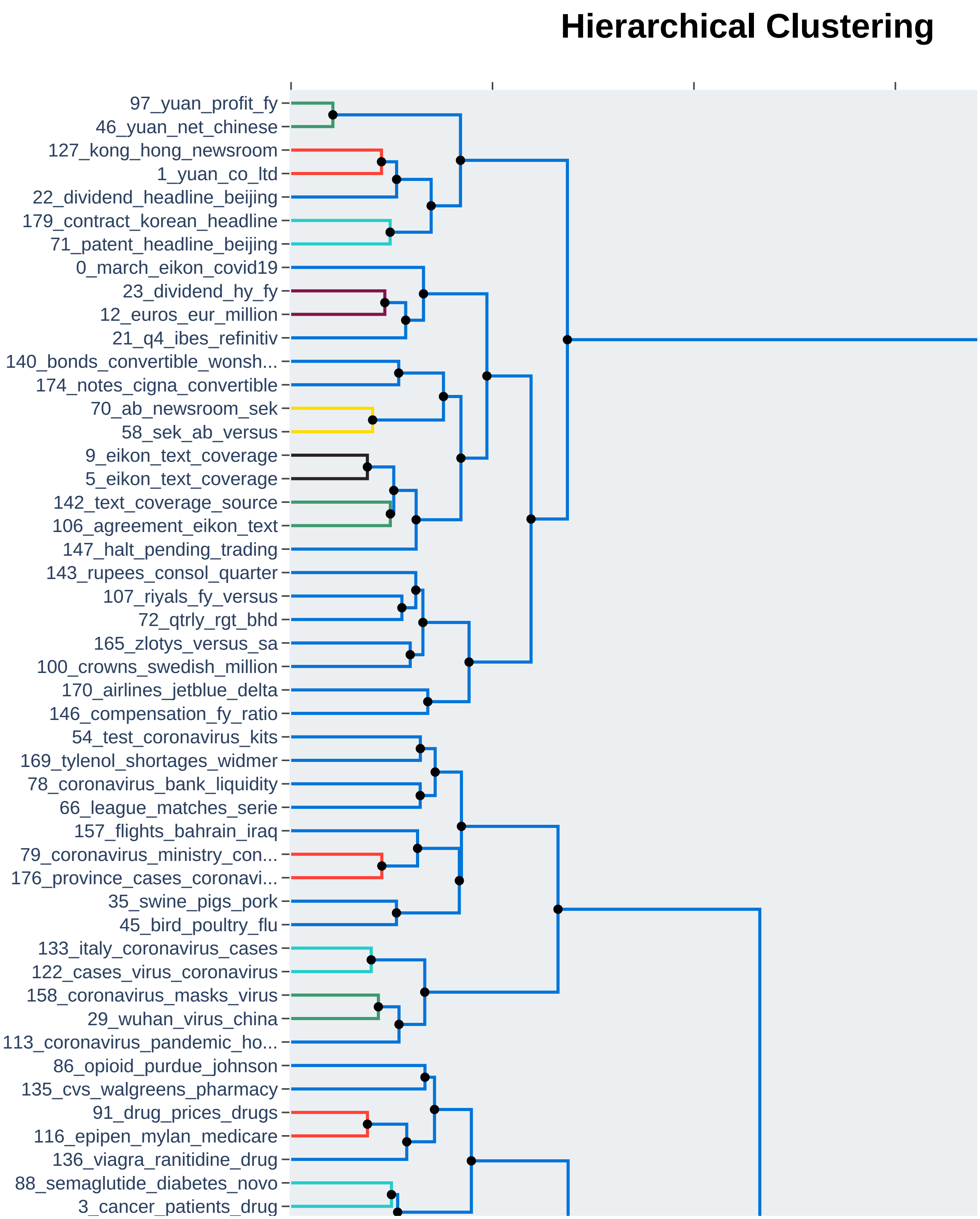
Hirarical Topics

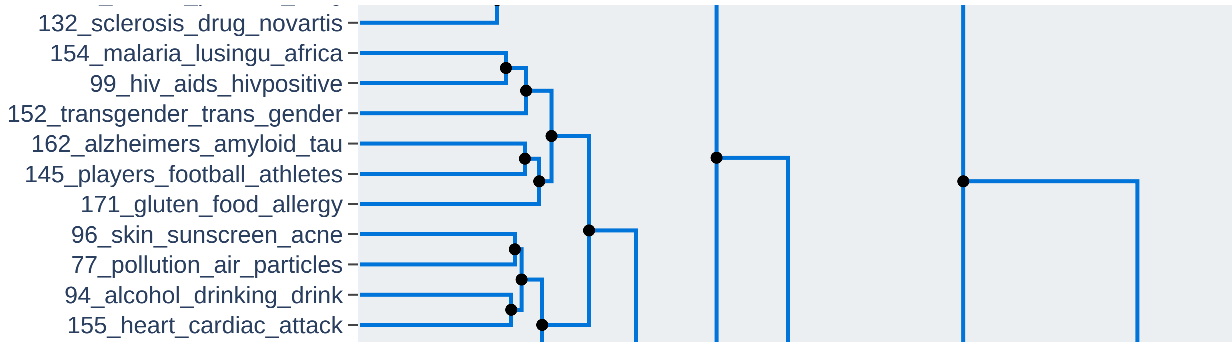
```
hierarchical_topics = model_health_tech.hierarchical_topics(docs_health_tech)
```

```
100%|██████████| 179/179 [00:02<00:00, 81.22it/s]
```

```
hierarchical_topics
```

```
model_health_tech.visualize_hierarchy(hierarchical_topics=hierarchical_topics)
```






```
tree = model_health_tech.get_topic_tree(hierarchical_topics)
print(tree)
```

```
.
├─text_coverage_source_eikon_million
│    ├─yuan_co_headline_ltd_beijing
│    │    ├─yuan_net_profit_fy_chinese
│    │    │    ├─■──yuan_profit_fy_net_million ── Topic: 97
│    │    │    └─■──yuan_net_chinese_yy_profit ── Topic: 46
│    │    └─headline_co_beijing_ltd_text
│    │         ├─co_headline_ltd_beijing_yuan
│    │         │    ├─co_yuan_ltd_text_headline
│    │         │    │    ├─■──kong_hong_newsroom_shares_ltd ── Topic: 127
│    │         │    │    └─■──yuan_co_ltd_headline_text ── Topic: 1
│    │         │    └─■──dividend_headline_beijing_shares_pay ── Topic: 22
│    │         └─patent_headline_korean_beijing_text
│    │              ├─■──contract_korean_headline_semiconductor_signed ── Top
│    │              └─■──patent_headline_beijing_text_source ── Topic: 71
│    └─eikon_coverage_text_source_million
│         ├─eikon_text_coverage_source_reuters
│         │    ├─eikon_text_coverage_eur_source
│         │    │    ├─■──march_eikon_covid19_text_coverage ── Topic: 0
│         │    │    └─euros_million_versus_eur_eikon
│         │    │         ├─euros_million_eur_versus_fy
│         │    │         │    ├─■──dividend_hy_fy_eikon_million ── Topic: 23
│         │    │         │    └─■──euros_eur_million_versus_ago ── Topic: 12
│         │    │         └─■──q4_ibes_refinitiv_2019_earnings ── Topic: 21
│         │    └─eikon_text_coverage_source_sek
│         │         ├─sek_ab_newsroom_eikon_gdynia
│         │         │    ├─bonds_convertible_notes_conversion_wonshare
│         │         │    │    ├─■──bonds_convertible_wonshare_conversion_head
│         │         │    │    └─■──notes_cigna_convertible_sek_senior ── Topi
│         │         │    └─sek_ab_newsroom_gdynia_eikon
│         │         │         ├─■──ab_newsroom_sek_gdynia_eikon ── Topic: 70
│         │         │         └─■──sek_ab_versus_million_loss ── Topic: 58
│         │         └─eikon_text_coverage_source_reuters
│         │              ├─eikon_text_coverage_source_reuters
│         │              │    ├─eikon_text_coverage_source_newsroom
│         │              │    │    ├─■──eikon_text_coverage_source_gdynia ──
│         │              │    │    └─■──eikon_text_coverage_source_appoints ─
│         │              │    └─agreement_text_eikon_coverage_source
│         │              │         ├─■──text_coverage_source_ltd_eikon ── Top
│         │              │         └─■──agreement_eikon_text_coverage_source
│         │              └─■──halt_pending_trading_ltd_eikon ── Topic: 147
│         │
```

```
│           └─versus_crowns_million_rupees_ago
│                ├─crowns_versus_million_rupees_ago
│                │    ├─rupees_qtrly_rgt_versus_riyals
│                │    │    ├─■──rupees_consol_quarter_dec_versus ── Topic: 143
│                │    │    └─qtrly_rgt_riyals_million_versus
│                │    │         ├─■──riyals_fy_versus_dirhams_ago ── Topic: 107
│                │    │         └─■──qtrly_rgt_bhd_million_profit ── Topic: 72
│                │    └─crowns_zlotys_versus_million_ago
│                │         ├─■──zlotys_versus_sa_ago_q1 ── Topic: 165
│                │         └─■──crowns_swedish_million_pct_norwegian ── Topic:
│                └─compensation_fy_ratio_ceo_inc
│                     ├─■──airlines_jetblue_delta_alaska_air ── Topic: 170
│                     └─■──compensation_fy_ratio_ceo_2019 ── Topic: 146
└─people_new_also_one_would
     ├─health_people_patients_study_disease
     │    ├─coronavirus_virus_outbreak_cases_china
     │    │    ├─coronavirus_swine_flu_poultry_bird
```

```
data_health_tech_sample['section'].unique()
```

```
array(['Technology News', 'health', 'Tech', 'tech', 'Healthcare',
       'Health', 'Technology, Media and Telecommunications',
       'Tech by VICE', 'Health News', 'Healthcare Facilities',
       'technology', 'Health and Science', 'HEALTH', 'Healthy Returns',
       'Health Insurance', 'Health ', 'Tech Guide',
       'Technology Executive Council', 'Tech Drivers', 'Health Care',
       'technology-quarterly', 'TECH', 'Tech Policy',
       'Technology, Media & Telecommunications',
       'health-care-ruling-gives-john-boehner-at-least-some-temporary-
vindication'],
      dtype=object)
```

```
topics_per_class = model_health_tech.topics_per_class(docs_health_tech, classes=da
```

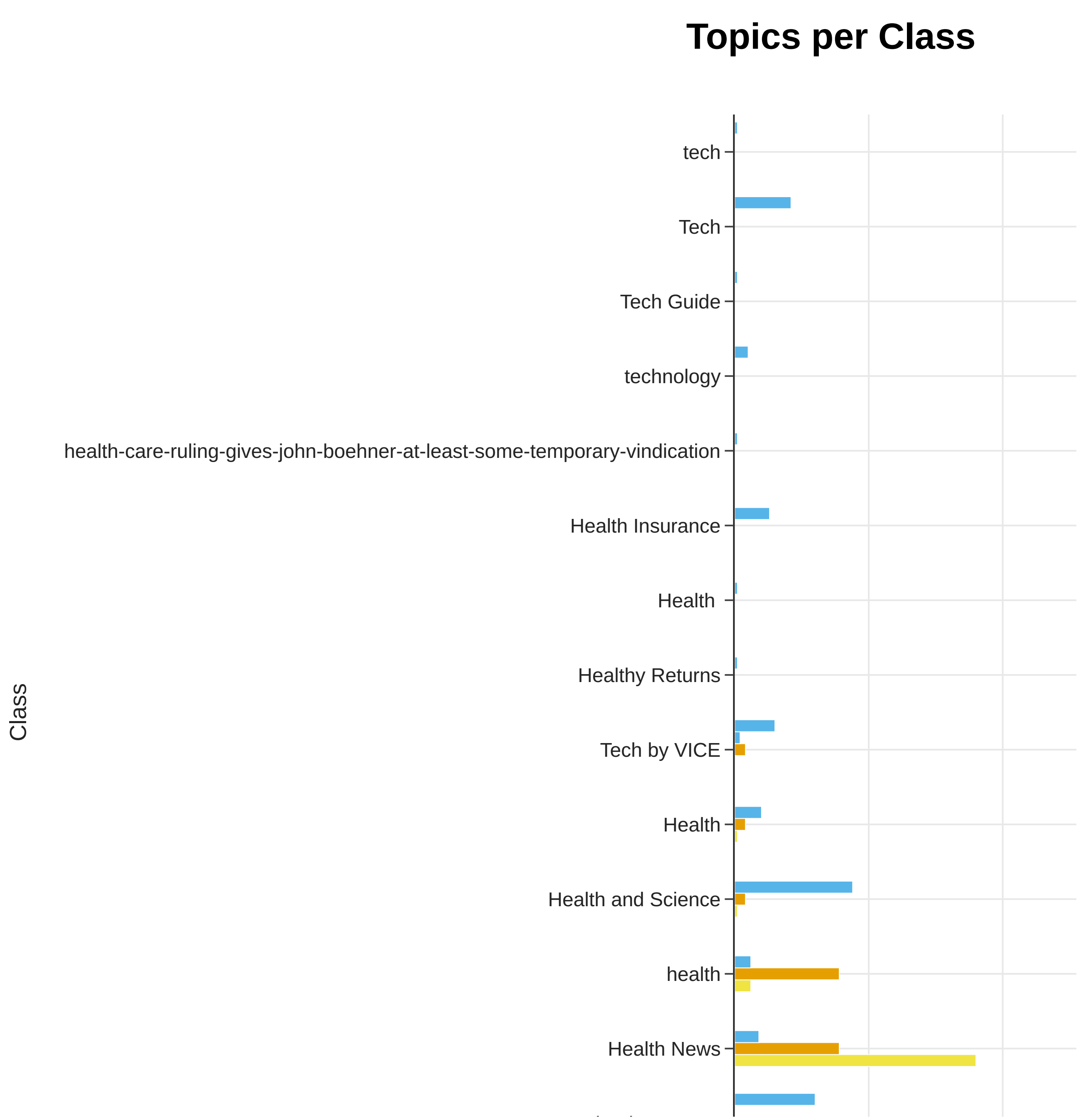
```
25it [00:07,  3.14it/s]
```

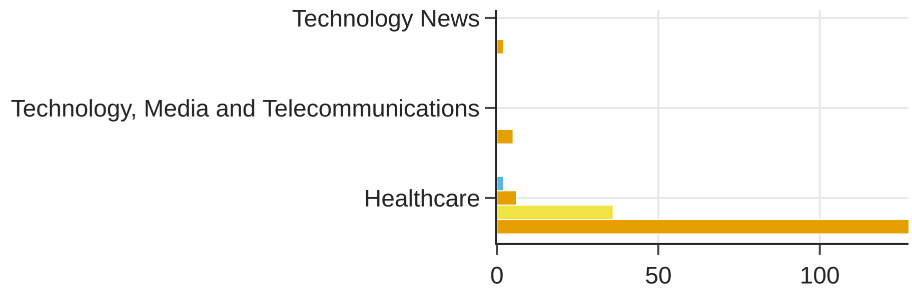
```
topics_per_class
```

	Topic	Words	Frequency	Class
0	8	care, hospital, medicaid, health, hospitals	1	Healthy Returns
1	74	crispr, gorsky, gene, orszag, returns	1	Healthy Returns
2	-1	lunch, sarscov2, virus, goldfish, its	9	Health
3	6	provape, ecigs, antivape, cigarettes, ecigarettes	1	Health
4	8	buttigieg, insurance, plan, premiums,	1	Health

		obamacare		
...	...	...	...	...
812	167	samsung, smartphone, trillion, samsungs, quarter	1	tech
813	168	robocalls, carriers, robocall, fcc, unwanted	1	tech
814	173	doordash, delivery, postmates, schaefer, dashers	2	tech

```
model_health_tech.visualize_topics_per_class(topics_per_class, top_n_topics=20)
```





Dynamic Modeling

- Analyzing the evolution of topics over time.
- How a topic is represented across different times.

```
#only extract month and year
timestamps = data_health_tech_sample['date'].to_list()
```

```
data_health_tech_sample['date']
```

```
0       2016-11-15 00:00:00
1       2019-07-23 15:33:55
2       2017-11-30 00:00:00
3       2019-08-09 00:00:00
4       2020-02-07 00:00:00
                ...
9995    2018-04-23 00:00:00
9996    2016-03-09 00:00:00
9997    2017-02-08 00:00:00
9998    2018-08-26 00:00:00
9999    2017-02-23 00:00:00
Name: date, Length: 10000, dtype: datetime64[ns]
```

```
len(data_health_tech_sample['date'].unique())
```

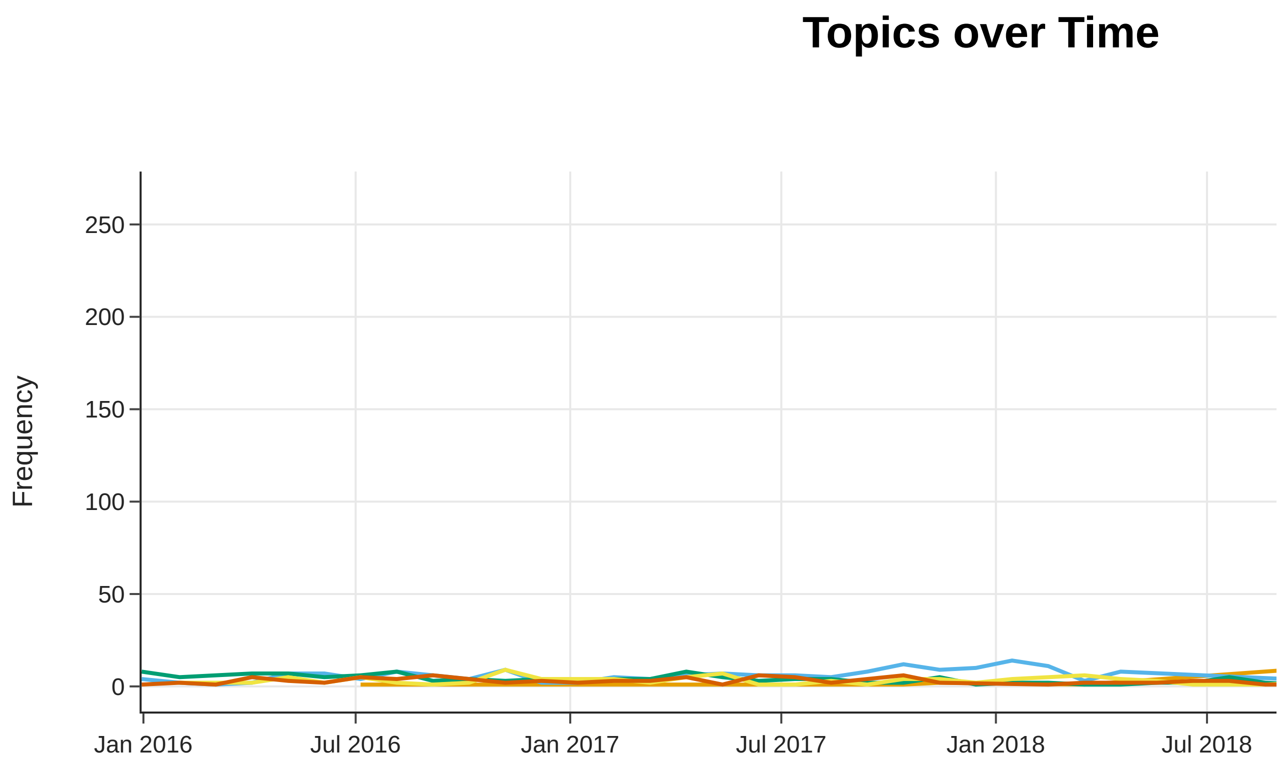
```
3689
```

```
topics_over_time = model_health_tech.topics_over_time(docs_health_tech, timestamps
```

```
50it [00:44,  1.11it/s]
```

Visualize Topic Over time

```
model_health_tech.visualize_topics_over_time(topics_over_time, top_n_topics=5)
```



Save Model

```
model_health_tech.save("GLG_V1_health_tech_topics_model")
```

Make Prediction

```
test_health_tech_data = data_tech_health.sample(n=10, random_state=3, ignore_index
```

```
test_health_tech_data = test_health_tech_data['article'].to_list()
```

```
topics_test, probs_test = model_health_tech.transform(test_health_tech_data)
```

```
Batches: 100% 1/1 [00:00<00:00, 1.16it/s]
2022-09-29 17:04:29,325 - BERTopic - Reduced dimensionality
2022-09-29 17:04:29,329 - BERTopic - Predicted clusters
```

```
topics_test
```

```
[-1, -1, 115, -1, 25, 18, 74, 34, -1, -1]
```

```
for i in topics_test:
  print(model_health_tech.get_topic(i))
```

```
[('people', 0.003683112979726207), ('new', 0.003526173610004055), ('one', 0.0
[('people', 0.003683112979726207), ('new', 0.003526173610004055), ('one', 0.0
[('cuomo', 0.047248323862186664), ('york', 0.032399770000899816), ('city', 0.
[('people', 0.003683112979726207), ('new', 0.003526173610004055), ('one', 0.0
[('amazon', 0.04918679022961324), ('grocery', 0.0263575100915574), ('foods',
[('cloud', 0.042000907741575715), ('revenue', 0.017953688965091277), ('micros
[('dna', 0.021598511624949475), ('genetic', 0.01941510047849327), ('fetal', 0
[('quantum', 0.03549049546117248), ('chips', 0.02614479028441985), ('computin
[('people', 0.003683112979726207), ('new', 0.003526173610004055), ('one', 0.0
[('people', 0.003683112979726207), ('new', 0.003526173610004055), ('one', 0.0
```

```
model_health_tech.get_topic(25)
```

```
[('amazon', 0.04918679022961324),
 ('grocery', 0.0263575100915574),
 ('foods', 0.02469608178880902),
 ('prime', 0.02463866534683422 5),
 ('delivery', 0.022764747053753687),
 ('stores', 0.0219881010141193),
 ('whole', 0.021039432998677474),
 ('amazons', 0.020034374753654705),
 ('shipping', 0.01595910074529306),
 ('walmart', 0.01556527276237221)]
```

Load Model

```
BerTopic_tech_model = BERTopic.load("GLG_V1_tech_topics_model")
```

```
BerTopic_health_model = BERTopic.load("GLG_V1_health_topics_model")
```

```
BerTopic_health_tech_model = BERTopic.load("GLG_V1_health_tech_topics_model")
```

References

- BERTopic: Neural topic modeling with a class-based TF-IDF procedure